**RAPPORT SAE BD**

**Sujet :**

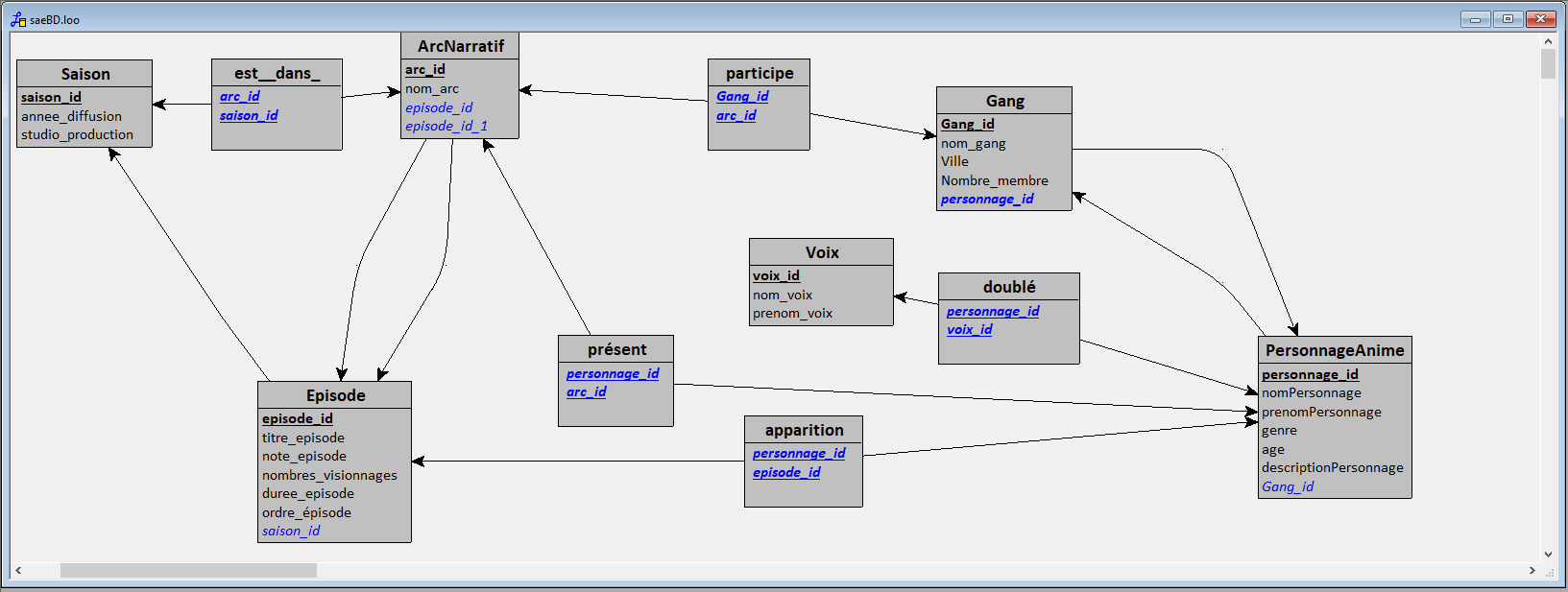
Nous avons pris comme sujet l’animé « Tokyo Revengers » qui est l’adaptation du manga de Ken Wakui. C’est un manga de type Shonen, Drame, Science-fiction et Romance qui parle d’un jeune adulte de 22 ans : Takemichi Hanagaki. Il a pour objectif de sauver sa petite amie assassinée en remontant le temps 12 ans en arrière et en rejoignant le gang responsable de sa mort dans le futur.

Pour ce qui est de la répartition des taches :

* Oumar : Ecriture des requêtes, création du SR, aide au MCD
* Saint-Nathan : Création du MCD, Création des Tables, Création des Vues, Création des Triggers et des Procédures.
* Milad : Création des INSERT

MCD :

SR :



**Création de tables :**

Voici le script de création des différentes tables du projet s’inspirant du MCD et du SR ci-dessus.

CREATE TABLE SAISON (

saison\_id INT PRIMARY KEY NOT NULL,

annee\_diffusion INT,

studio\_production VARCHAR(255)

);

CREATE TABLE EPISODE (

episode\_id INT PRIMARY KEY NOT NULL,

titre\_episode VARCHAR(255),

note\_episode INT,

nombres\_visionnages INT,

duree\_episode INT,

Ordre\_episode INT,

saison\_id INT,

FOREIGN KEY (saison\_id ) REFERENCES SAISON(saison\_id )

);

CREATE TABLE ArcNarratif (

arc\_id INT PRIMARY KEY NOT NULL,

nom\_arc VARCHAR(255),

id\_episode\_debut INT,

id\_episode\_fin INT,

episode\_id INT,

FOREIGN KEY (id\_episode\_debut) REFERENCES EPISODE(episode\_id),

FOREIGN KEY (id\_episode\_fin) REFERENCES EPISODE(episode\_id)

);

CREATE TABLE Est\_dans\_ (

arc\_id INT,

saison\_id INT,

PRIMARY KEY (arc\_id,saison\_id),

FOREIGN KEY (saison\_id ) REFERENCES SAISON(saison\_id ),

FOREIGN KEY (arc\_id) REFERENCES ArcNarratif (arc\_id)

);

CREATE TABLE Voix (

voix INT PRIMARY KEY NOT NULL,

nom\_voix VARCHAR(255),

prenom\_voix VARCHAR(255)

);

CREATE TABLE Gang\_1(

Gang\_id INT PRIMARY KEY NOT NULL,

nom\_Gang VARCHAR(255),

PersonnageAnime\_id INT

-- FOREIGN KEY (PersonnageAnime\_id ) REFERENCES PersonnageAnime(PersonnageAnime\_id )

);

CREATE TABLE Participe\_1(

Gang\_id INT,

arc\_id INT,

PRIMARY KEY (Gang\_id,arc\_id),

FOREIGN KEY (Gang\_id) REFERENCES Gang\_1(Gang\_id),

FOREIGN KEY (arc\_id) REFERENCES ArcNarratif (arc\_id)

);

CREATE TABLE PersonnageAnime\_1(

PersonnageAnime\_id INT PRIMARY KEY NOT NULL,

nomPersonnage VARCHAR(255),

prenomPersonnage VARCHAR(255),

genre VARCHAR(55),

age INT,

descriptionPersonnage VARCHAR(255),

Gang\_id INT,

FOREIGN KEY (Gang\_id) REFERENCES Gang\_1(Gang\_id)

);

CREATE TABLE double\_1(

PersonnageAnime\_id INT,

voix INT,

PRIMARY KEY (PersonnageAnime\_id,voix),

FOREIGN KEY (PersonnageAnime\_id ) REFERENCES PersonnageAnime\_1(PersonnageAnime\_id ),

FOREIGN KEY (voix) REFERENCES Voix(voix)

);

CREATE TABLE Apparition\_1 (

PersonnageAnime\_id INT,

episode\_id INT,

PRIMARY KEY (PersonnageAnime\_id,episode\_id),

FOREIGN KEY (PersonnageAnime\_id ) REFERENCES PersonnageAnime\_1(PersonnageAnime\_id ),

FOREIGN KEY (episode\_id) REFERENCES EPISODE(episode\_id)

);

CREATE TABLE est\_present\_1(

PersonnageAnime\_id INT,

arc\_id INT,

PRIMARY KEY (PersonnageAnime\_id,arc\_id),

FOREIGN KEY (PersonnageAnime\_id ) REFERENCES PersonnageAnime\_1(PersonnageAnime\_id ),

FOREIGN KEY (arc\_id) REFERENCES ArcNarratif (arc\_id)

);

--ALTER TABLE Gang\_1 MODIFY PersonnageAnime\_id REFERENCES PersonnageAnime\_1(PersonnageAnime\_id );

Nous n’avons pas vraiment mis de contraintes car de nombreux éléments pouvaient être NULL, il n’ y a pas d’âge minimun ou limite dans la série.

**Les INSERT :**

Voici une partie des différents script d’insertions toutes écrites à la main.

INSERT INTO SAISON (saison\_id, annee\_diffusion, studio\_diffusion)

VALUES (1, 2021, 'Liden Films');

INSERT INTO SAISON (saison\_id, annee\_diffusion, studio\_diffusion)

VALUES (2, 2023, 'Liden Films');

INSERT INTO Gang\_1 (gang\_id, nom\_gang, ville, nombre\_membre, personnage\_id)

VALUES (1, 'Tokyo Manjikai', 'Tokyo (Shibuya)', 100, 2);

INSERT INTO Gang\_1 (gang\_id, nom\_gang, ville, nombre\_membre, personnage\_id)

VALUES (2, 'Mœbius', 'Tokyo (Shinjuku)', 50, 20);

INSERT INTO Gang\_1 (gang\_id, nom\_gang, ville, nombre\_membre, personnage\_id)

VALUES (3, 'Valhalla', 'Tokyo (Shibuya)', 300, 13);

INSERT INTO Gang\_1 (gang\_id, nom\_gang, ville, nombre\_membre, personnage\_id)

VALUES (4, 'Black Dragons', 'Tokyo (Shibuya)', 100, 21);

INSERT INTO ArcNarratif (arc\_id, nom\_arc, episode\_id, episode\_id\_1)

VALUES (1, 'arc Toman', 1, 5);

INSERT INTO ArcNarratif (arc\_id, nom\_arc, episode\_id, episode\_id\_1)

VALUES (2, 'arc Mœbius', 6, 7);

INSERT INTO ArcNarratif (arc\_id, nom\_arc, episode\_id, episode\_id\_1)

VALUES (3, 'arc Sanglant D Halloween (Valhalla)', 13, 12);

INSERT INTO ArcNarratif (arc\_id, nom\_arc, episode\_id, episode\_id\_1)

VALUES (4, 'Arc De Confrontation De Noël (Black Dragons)', 25, 12);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES (1, 'Takemichi', 'Hanagaki', 'Masculin', 26, 'personnage principal, c’est un jeune homme de 26 ans qui apprend par hasard qu’ il peut voyager dans le temps de 12 ans en arrière jour pour jour. Il utilise originellement ce pouvoir pour secourir son ancienne petite amie Tachibana Hinata', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(2, 'Sano', 'Manjiro', 'Masculin', 15, 'Il est l’un des membres fondateurs et le leader du gang Tokyo Manjikai', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(3, 'Ryugoji', 'Ken', 'Masculin', 15, 'Il officie en tant que numéro 2 du Tokyo manjikai', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(4, 'Tachibana', 'Hinata', 'Féminin', 26, 'elle était la petite amie d Hanagaki Takemichi au collège', NULL);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(5, 'Tachibana', 'Naoto', 'Masculin', 25, 'frère de Tachibana Hinata. Afin de sauver celle-ci, il est devenu lieutenant de police et s est allié avec Hanagaki Takemichi', NULL);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(6, 'Yamagishi', 'Kazushi', 'Masculin', 14, 'il était un élève du collège Mizo, il a depuis rejoint le gang Tokyo Manjikai', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(7, 'Suzuki', 'Makoto', 'Masculin', 14, 'il était un élève du collège Mizo, il a depuis rejoint le gang Tokyo Manjikai', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(8, 'Yamamoto', 'Takuya', 'Masculin', 14, 'il était un élève du collège Mizo, il a depuis rejoint le gang Tokyo Manjikai', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(9, 'Sendo', 'Atsushi', 'Masculin', 14, 'il était un élève du collège Mizo, il a depuis rejoint le gang Tokyo Manjikai', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(10, 'Masaru', NULL, 'Masculin', 15, 'le cousin à Takemichi', NULL);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(11, 'Akaishi', 'Jin', 'Masculin', 15, 'est un voyou membre du Gang de Kiyomasa et de la troisième division du Tokyo Manjikai', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(12, 'Kiyomizu', 'Masataka', 'Masculin', 14, 'alias Kiyomasa est un voyou membre de la troisième division du Tokyo Manjikai', 1);

INSERT INTO PersonnageAnime\_1 (PersonnageAnime\_id, nomPersonnage, prenomPersonnage, genre, age, descriptionPersonnage, Gang\_id)

VALUES(13, 'Kisaki', 'Tetta', 'Masculin', 13, 'c est le principal antagoniste de Tokyo Revengers. Il est connu notamment pour avoir gravi à une vitesse folle les échelons du Toman et pour avoir organisé à de nombreuses reprises le meurtre de Tachibana Hinata', 3);

**Requêtes :**

Voici une parties des requêtes répondant à des questions posées sur la base de données de Tokyo Revengers. ( fichier Réponse et fichier réponse 2 )

Quel est le nombre total de visionnages pour chaque saison de l'anime "Tokyo Revengers" ?

SELECT S.saison\_id, S.annee\_diffusion, SUM(E.nombres\_visionnages) AS total\_visionnages

FROM Saison S

JOIN Episode E ON S.saison\_id = E.saison\_id

GROUP BY S.saison\_id, S.annee\_diffusion;

Quel est le nombre d'épisodes dans la saison avec le plus grand nombre de visionnages ?

SELECT saison\_id, COUNT(\*) AS nombre\_episodes

FROM Episode

WHERE saison\_id = (

SELECT saison\_id

FROM Episode

GROUP BY saison\_id

ORDER BY SUM(nombres\_visionnages) DESC

LIMIT 1

)

GROUP BY saison\_id;

Quels sont les personnages qui ont participé à des épisodes ayant une note supérieure à 8 ?

SELECT PA.nomPersonnage, PA.prenomPersonnage

FROM PersonnageAnime PA

JOIN Apparition A ON PA.PersonnageAnime\_id = A.PersonnageAnime\_id

JOIN Episode E ON A.episode\_id = E.episode\_id

WHERE E.note\_episode > 8;

Quel est le nombre moyen de visionnages par épisode pour chaque gang de l'anime "Tokyo Revengers" ?

SELECT G.nom\_Gang, AVG(E.nombres\_visionnages) AS nombre\_moyen\_visionnages

FROM Gang G

JOIN PersonnageAnime PA ON G.Gang\_id = PA.Gang\_id

JOIN Apparition A ON PA.PersonnageAnime\_id = A.PersonnageAnime\_id

JOIN Episode E ON A.episode\_id = E.episode\_id

GROUP BY G.Gang\_id, G.nom\_Gang;

Quel est le personnage qui a la plus haute note moyenne pour ses apparitions dans les épisodes ?

SELECT PA.nomPersonnage, PA.prenomPersonnage

FROM PersonnageAnime PA

WHERE PA.PersonnageAnime\_id = (

SELECT A.PersonnageAnime\_id

FROM Apparition A

JOIN Episode E ON A.episode\_id = E.episode\_id

GROUP BY A.PersonnageAnime\_id

ORDER BY AVG(E.note\_episode) DESC

LIMIT 1

);

Quels sont les arcs narratifs qui ont des épisodes avec une note moyenne supérieure à 4 ?

SELECT AN.nom\_arc

FROM ArcNarratif AN

JOIN Episode E ON AN.arc\_id = E.arc\_id

GROUP BY AN.arc\_id, AN.nom\_arc

HAVING AVG(E.note\_episode) > ;

Quels sont les personnages qui n'ont jamais été visionnés dans un épisode ?

SELECT PA.nomPersonnage, PA.prenomPersonnage

FROM PersonnageAnime PA

WHERE PA.PersonnageAnime\_id NOT IN (

SELECT DISTINCT A.PersonnageAnime\_id

FROM Apparition A

);

Quelle est la saison avec le plus grand nombre d'épisodes impliquant un personnage spécifique ?

SELECT saison\_id, COUNT(\*) AS nombre\_episodes

FROM Apparition

WHERE PersonnageAnime\_id = (SELECT PersonnageAnime\_id FROM PersonnageAnime WHERE nomPersonnage = 'Nom du personnage')

GROUP BY saison\_id

ORDER BY COUNT(\*) DESC

LIMIT 1;

Quel est le personnage qui a joué le plus grand nombre d'épisodes dans un arc narratif donné ?

SELECT PA.nomPersonnage, PA.prenomPersonnage

FROM PersonnageAnime PA

WHERE PA.PersonnageAnime\_id = (

SELECT A.PersonnageAnime\_id

FROM Apparition A

JOIN Episode E ON A.episode\_id = E.episode\_id

WHERE A.arc\_id = (SELECT arc\_id FROM ArcNarratif WHERE nom\_arc = "Nom de l'arc narratif")

GROUP BY A.PersonnageAnime\_id

ORDER BY COUNT(\*) DESC

LIMIT 1

);

Quelle est la voix du personnage principal dans l'anime "Tokyo Revengers" ?

SELECT V.nom\_voix, V.prenom\_voix

FROM Voix V

JOIN Double D ON V.voix = D.voix

JOIN PersonnageAnime PA ON D.PersonnageAnime\_id = PA.PersonnageAnime\_id

WHERE PA.nomPersonnage = 'Hanagaki';

Quels personnages sont apparus dans plus d'un arc ?

SELECT PA.nomPersonnage, PA.prenomPersonnage

FROM PersonnageAnime PA

JOIN Apparition A ON PA.PersonnageAnime\_id = A.PersonnageAnime\_id

JOIN ArcNarratif AN ON A.arc\_id = AN.arc\_id

GROUP BY PA.PersonnageAnime\_id, PA.nomPersonnage, PA.prenomPersonnage

HAVING COUNT(DISTINCT AN.arc\_id) > 1;

Trier les gangs par taille de leur nombre de membres dans l'ordre croissant.

SELECT G.Gang\_id, G.nom\_Gang, COUNT(PA.PersonnageAnime\_id) AS nombre\_membres

FROM Gang G

LEFT JOIN PersonnageAnime PA ON G.Gang\_id = PA.Gang\_id

GROUP BY G.Gang\_id, G.nom\_Gang

ORDER BY COUNT(PA.PersonnageAnime\_id) ASC;

Quels sont les épisodes où le personnage principal n'apparaît pas ?

SELECT E.titre\_episode

FROM Episode E

WHERE E.episode\_id NOT IN (

SELECT A.episode\_id

FROM Apparition A

JOIN PersonnageAnime PA ON A.PersonnageAnime\_id = PA.PersonnageAnime\_id

WHERE PA.nomPersonnage = 'Hanagaki'

);

Les Vues :

Voici les vues que nous avons créer pour cette base de données.

--Vue des épisodes avec leurs informations de saison

CREATE VIEW Vue\_Episodes AS

SELECT E.episode\_id, E.titre\_episode, E.note\_episode, E.nombres\_visionnages, E.duree\_episode, E.Ordre\_episode, S.saison\_id, S.annee\_diffusion, S.studio\_production

FROM EPISODE E

JOIN SAISON S ON E.saison\_id = S.saison\_id;

--Vue des arcs narratifs avec les noms des épisodes de début et de fin :

CREATE VIEW Vue\_ArcNarratif AS

SELECT A.arc\_id, A.nom\_arc, Edebut.titre\_episode AS episode\_debut, Efin.titre\_episode AS episode\_fin

FROM ArcNarratif A

JOIN EPISODE E ON A.id\_episode\_debut = E.episode\_id

JOIN EPISODE E ON A.id\_episode\_fin = E.episode\_id;

--Vue des personnages avec leurs informations de voix et de gang :

CREATE VIEW Vue\_Personnage AS

SELECT P.PersonnageAnime\_id, P.nomPersonnage, P.prenomPersonnage, P.genre, P.age, P.descriptionPersonnage, V.nom\_voix, V.prenom\_voix, G.nom\_Gang

FROM PersonnageAnime P

JOIN Voix V ON P.PersonnageAnime\_id = V.voix

JOIN Gang G ON P.Gang\_id = G.Gang\_id;

Les Triggers :

Voici différents trigger que nous avons réaliser pour ce projet.

--Trigger pour mettre à jour le nombre de membres d'un gang lorsque des personnages y sont ajoutés ou supprimés :

CREATE TRIGGER MiseAJourNombreMembres

AFTER INSERT OR DELETE ON PersonnageAnime

FOR EACH ROW

BEGIN

DECLARE gangId INT;

SET gangId = NEW.Gang\_id;

IF gangId IS NOT NULL THEN

UPDATE Gang

SET nombre\_membres = (SELECT COUNT(\*) FROM PersonnageAnime WHERE Gang\_id = gangId)

WHERE Gang\_id = gangId;

END IF;

END;

--Trigger pour empêcher la suppression d'une saison ayant des épisodes non terminés :

CREATE TRIGGER EmpecherSuppressionSaison

BEFORE DELETE ON Saison FOR EACH ROW

BEGIN

DECLARE episodeNonTermine INT;

SELECT COUNT(\*) INTO episodeNonTermine

FROM Episode

WHERE saison\_id = OLD.saison\_id AND id\_episode\_fin IS NULL;

IF episodeNonTermine > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Impossible de supprimer la saison. Des épisodes sont en cours.';

END IF;

END;

--Trigger pour mettre à jour la durée totale d'un arc narratif lorsque des épisodes y sont ajoutés ou supprimés :

CREATE TRIGGER MiseAJourDureeArcNarratif

AFTER INSERT OR DELETE ON Episode FOR EACH ROW

BEGIN

DECLARE arcId INT;

SET arcId = NEW.arc\_id;

IF arcId IS NOT NULL THEN

UPDATE ArcNarratif

SET duree\_totale = (SELECT SUM(duree\_episode) FROM Episode WHERE arc\_id = arcId)

WHERE arc\_id = arcId;

END IF;

END;

--Trigger pour mettre à jour le nombre moyen de visionnages par épisode d'un gang lorsque des personnages y sont ajoutés ou supprimés :

CREATE TRIGGER MiseAJourNombreMoyenVisionnagesParGang

AFTER INSERT OR DELETE ON PersonnageAnime FOR EACH ROW

BEGIN

DECLARE gangId INT;

SET gangId = NEW.Gang\_id;

IF gangId IS NOT NULL THEN

UPDATE Gang

SET nombre\_moyen\_visionnages = (SELECT AVG(E.nombres\_visionnages)

FROM Episode E

JOIN Apparition A ON E.episode\_id = A.episode\_id

JOIN PersonnageAnime PA ON A.PersonnageAnime\_id = PA.PersonnageAnime\_id

WHERE PA.Gang\_id = gangId)

WHERE Gang\_id = gangId;

END IF;

END;

--Trigger pour mettre à jour la liste des personnages qui ont participé à des épisodes avec une note supérieure à 4 :

CREATE TRIGGER MiseAJourPersonnagesNoteSuperieure

AFTER INSERT OR UPDATE OF note\_episode ON Episode FOR EACH ROW

BEGIN

IF NEW.note\_episode > 4 THEN

INSERT INTO PersonnagesNoteSuperieure (PersonnageAnime\_id)

VALUES (NEW.PersonnageAnime\_id)

ON DUPLICATE KEY UPDATE PersonnageAnime\_id = NEW.PersonnageAnime\_id;

ELSE

DELETE FROM PersonnagesNoteSuperieure

WHERE PersonnageAnime\_id = NEW.PersonnageAnime\_id;

END IF;

END;

--Trigger pour mettre à jour la liste des arcs narratifs ayant des épisodes avec une note moyenne supérieure à 4 :

CREATE TRIGGER MiseAJourArcsNoteMoyenneSuperieure

AFTER INSERT OR UPDATE OF note\_episode ON Episode FOR EACH ROW

BEGIN

IF NEW.note\_episode > 4 THEN

INSERT INTO ArcsNoteMoyenneSuperieure (arc\_id)

VALUES (NEW.arc\_id)

ON DUPLICATE KEY UPDATE arc\_id = NEW.arc\_id;

ELSE

DELETE FROM ArcsNoteMoyenneSuperieure

WHERE arc\_id = NEW.arc\_id;

END IF;

END;

--Trigger pour mettre à jour le nombre d'épisodes impliquant un personnage spécifique dans une saison :

CREATE TRIGGER MiseAJourNombreEpisodesPersonnageSaison

AFTER INSERT OR DELETE ON Apparition FOR EACH ROW

BEGIN

DECLARE personnageId INT;

DECLARE saisonId INT;

SET personnageId = NEW.PersonnageAnime\_id;

SET saisonId = NEW.saison\_id;

IF personnageId IS NOT NULL AND saisonId IS NOT NULL THEN

UPDATE Saison

SET nombre\_episodes\_personnage = (SELECT COUNT(\*)

FROM Apparition

WHERE PersonnageAnime\_id = personnageId AND saison\_id = saisonId)

WHERE saison\_id = saisonId;

END IF;

END;

--Trigger pour mettre à jour le nombre d'épisodes dans la saison avec le plus grand nombre de visionnages :

CREATE TRIGGER MiseAJourSaisonPlusVisionnees

AFTER INSERT ON Episode FOR EACH ROW

BEGIN

DECLARE saisonId INT;

SET saisonId = NEW.saison\_id;

IF saisonId IS NOT NULL THEN

UPDATE Saison

SET nombre\_episodes\_plus\_visionnes = (SELECT COUNT(\*)

FROM Episode

WHERE saison\_id = saisonId

GROUP BY saison\_id

ORDER BY SUM(nombres\_visionnages) DESC

LIMIT 1)

WHERE saison\_id = saisonId;

END IF;

END;