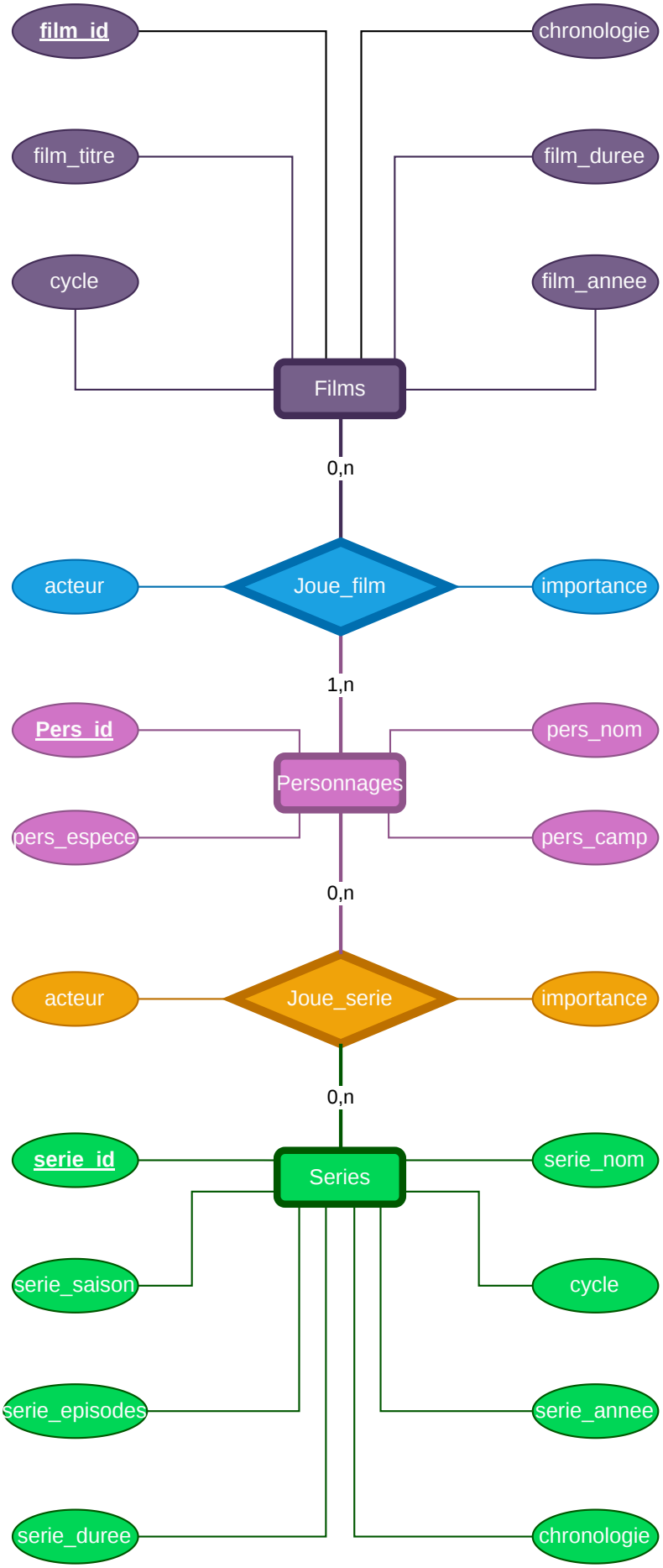
G4 Projet SGBD : Etape 2 Univers Star Wars

L2 Informatique

I. Rappel du Modèle Entité-Association

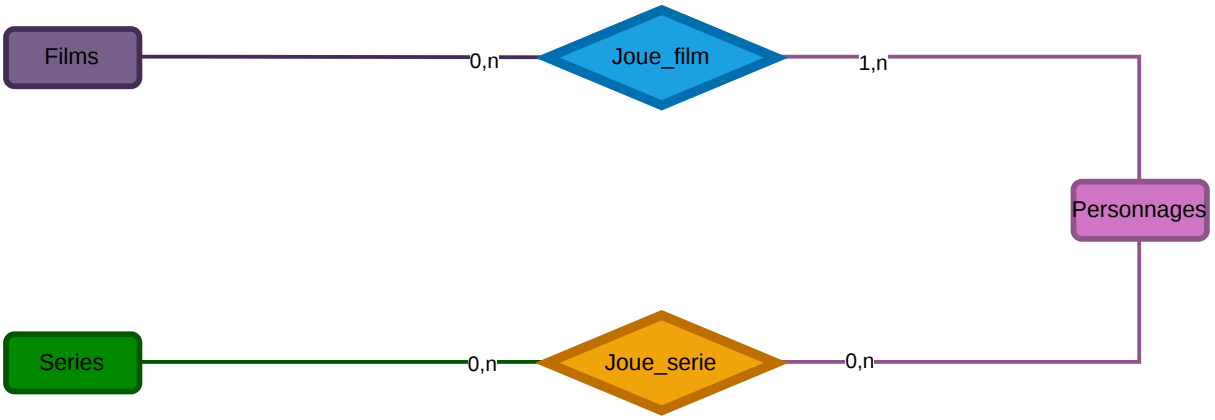
I.1 Schéma E/A (avec attributs)


film_id
chronologie
film_titre
film_duree
cycle
film_annee
Films
0,n
acteur
Joue_film
importance
1,n
Pers_id
pers_nom
Personnages
pers_espece
pers_camp
0,n
acteur
Joue_serie
importance
0,n
serie_id
serie_nom
Series
serie_saison
cycle
serie_episodes
serie_annee
serie_duree
chronologie

Entités : Films, Series, Personnages

Associations : Joue_film (n:m), Joue_serie (n:m)

I.2 Schéma E/A simplifié



Entités : Films, Series, Personnages

Associations : Joue_film (n:m), Joue_serie (n:m)

II. Algorithme de Transformation E/A → Relationnel

II.1 Règles générales de transformation

L'algorithme de transformation d'un modèle Entité-Association vers un modèle relationnel suit les règles suivantes :

Règle 1 : Transformation des entités en relation

Principe : Chaque entité devient une table (relation) qui reprend ses attributs et sa clé primaire.

Exemple :

- Entité Films → Relation Films (film_id, film_titre, ...)

Règle 2 : Transformation des associations n:m

Principe : Une association n:m devient une table dont la clé primaire est composée des clés des entités associées.

Elle garde aussi ses attributs propres.

Exemple :

- Association Joue_film → Relation Joue_film (film_id, pers_id, importance,...)

Règle 3 : Transformation des associations 1:n

Principe : La clé primaire du côté “1” devient clé étrangère dans la relation du côté “n”.

Exemple :

- Dans notre modèle, seule Joue_film contient une cardinalité minimale à 1 du côté Personnages, mais cela ne change pas la transformation car l’association reste n:m.

Règle 4 : Gestion des cardinalités minimales

Principe :

Les cardinalités minimales deviennent des contraintes :

- **(1,n)** → clé étrangère **NOT NULL**
- **(0,n)** → clé étrangère **NULL possible**

Exemple :

- Films (0,n) → Joue_film : film_id peut apparaître ou non dans Joue_film.
- Personnages (1,n) → Joue_Film : un personnage (`pers_id`) doit apparaître dans au moins un film → NOT NULL
- Dans Joue_serie, toutes les participations sont **optionnelles** des deux côtés → clés étrangères **NULL possibles**.

II.2 Étapes de l'algorithme appliqué

Étape 1 : Identifier toutes les entités

→ Films, Series, Personnages

Étape 2 : Transformer chaque entité en relation

→ Création des tables Films, Series, Personnages

Étape 3 : Identifier les associations et leur type

→ Joue_film (n:m), Joue_serie (n:m)

Étape 4 : Transformer les associations n:m en relations

→ Création des tables Joue_film et Joue_serie avec les clés et leurs attributs.

Étape 5 : Définir les contraintes d'intégrité

→ PRIMARY KEY, FOREIGN KEY, NOT NULL, CHECK

III. Application à notre Modèle

III.1 Transformation de l'entité Films

Entité E/A :

Films

- `film_id (PK)`
- `film_titre`
- `film_duree`
- `film_annee`
- `cycle`
- `chronologie`

Relation obtenue :

```
G04_Films (
    film_id INT PRIMARY KEY,
    film_titre VARCHAR(200) NOT NULL,
    film_duree INT,
    film_annee INT NOT NULL,
    cycle VARCHAR(100),
    chronologie VARCHAR(20)
)
```

Explication :

- `film_id` devient la clé primaire (contrainte PRIMARY KEY)
- `film_titre` et `film_annee` sont obligatoires (NOT NULL)
- Les autres attributs peuvent être NULL (optionnels)

III.2 Transformation de l'entité Series

Entité E/A :

Series

- serie_id (PK)
- serie_nom
- serie_duree
- serie_annee
- serie_saison
- cycle
- chronologie

Relation obtenue :

```
G04_Series (
    serie_id INT PRIMARY KEY,
    serie_nom VARCHAR(200) NOT NULL,
    serie_duree INT,
    serie_annee INT NOT NULL,
    serie_saisons INT,
    cycle VARCHAR(100),
    chronologie VARCHAR(20)
)
```

III.3 Transformation de l'entité Personnages

Entité E/A :

Personnages

- pers_id (PK)
- pers_nom
- pers_espece
- pers_camp

Relation obtenue :

```
G04_Personnages (
    pers_id INT PRIMARY KEY,
    pers_nom VARCHAR(100) NOT NULL,
    pers_espece VARCHAR(50),
    pers_camp VARCHAR(50)
)
```

III.4 Transformation de l'association Joue_film

Association E/A :

Joue_film (n:m)

- Liaison des tables : Films ↔ Personnages
- Clés primaires : `(film_id, pers_id)`
- Attributs : `importance`, `acteur`
- Ajout des clés étrangères : `film_id` (→ Films), `pers_id` (→ Personnages)

Relation obtenue :

```
G04_Joue_film (
    film_id INT,
    pers_id INT,
    importance VARCHAR(10) NOT NULL CHECK (importance IN ('Principal', 'Secondaire')),
    acteur VARCHAR(100),
    PRIMARY KEY (film_id, pers_id),
    FOREIGN KEY (film_id) REFERENCES Films(film_id),
    FOREIGN KEY (pers_id) REFERENCES Personnages(pers_id)
)
```

III.5 Transformation de l'association Joue_serie

Association E/A :

Joue_serie (n:m)

- Liaison des tables : Series ↔ Personnages

- Clés primaires : `(serie_id, pers_id)`
- Attributs : `importance`, `acteur`
- Ajout des clés étrangères : `serie_id` (→ Serie), `pers_id` (→ Personnages)

Relation obtenue :

```
G04_Joue_serie (
    serie_id INT,
    pers_id INT,
    importance VARCHAR(10) NOT NULL CHECK (importance IN ('Principal', 'Secondaire')),
    acteur VARCHAR(100),
    PRIMARY KEY (serie_id, pers_id),
    FOREIGN KEY (serie_id) REFERENCES Series(serie_id),
    FOREIGN KEY (pers_id) REFERENCES Personnages(pers_id)
)
```

IV. Modèle Relationnel

IV.1 Schéma relationnel

Précisions :

- Clé primaire : soulignée
- Clé étrangère : préfixée par #

G04_Films(**film_id** : INT, film_titre : VARCHAR(200), film_duree : INT, film_annee : INT, cycle : VARCHAR(100), chronologie : VARCHAR(20))

G04_Series(**serie_id** : INT, serie_nom : VARCHAR(200), serie_duree : INT, serie_annee : INT, serie_saisons : INT, cycle : VARCHAR(100), chronologie : VARCHAR(20))

G04_Personnages(**pers_id** : INT, pers_nom : VARCHAR(100), pers_espece : VARCHAR(50), pers_camp : VARCHAR(50))

G04_Joue_film(#**film_id** : INT, #**pers_id** : INT, importance : VARCHAR(10) NOT NULL CHECK (G04_Joue_film ('Principal', 'Secondaire'), acteur : VARCHAR(100))

- Clé primaire : (film_id, pers_id)
- Clés étrangères : film_id → Films(film_id), pers_id → Personnages(pers_id)

G04_Joue_serie(#**serie_id** : INT, #**pers_id** : INT, importance : VARCHAR(10) NOT NULL CHECK (G04_Joue_serie IN ('Principal', 'Secondaire'), acteur : VARCHAR(100))

- Clé primaire : (serie_id, pers_id)
- Clés étrangères : serie_id → Series(serie_id), pers_id → Personnages(pers_id)

V. Contraintes d'intégrité

V.1 Contraintes de clés

Clés primaires :

- Garantissent l'unicité de chaque ligne
- Ne peuvent pas être NULL
- Tables : Films, Series, Personnages (clés simples)
- Tables : Joue_film, Joue_serie (clés primaires)

Clés étrangères :

- Permet de garantir une bonne integrité
- Empêchent les valeurs qui ne sont pas associé à une autre table

---

V.2 Contraintes de valeurs

Contraintes de valeurs non nulle dans certains attributs avec NOT NULL :

```
-- Mettre obligatoirement du contenu
film_titre NOT NULL
film_annee NOT NULL
serie_nom NOT NULL
serie_annee NOT NULL
pers_nom NOT NULL
```

Contraintes de cohérence dans les valeurs avec `CHECK` :

```
-- Années de sortie cohérentes
CHECK (film_annee >= 1977)  -- Premier Star Wars en 1977
CHECK (serie_annee >= 1977)

-- Durées positives
CHECK (film_duree > 0) -- Une duree qui doit être supérieur à 0 min
CHECK (serie_duree > 0)
CHECK (serie_saisons > 0) -- Il doit y avoir au moins une saison à une série.
```

- `film_annee`, `serie_annee` doivent être ≥ 1977 (année du premier Star Wars)
- `film_duree`, `serie_duree` doivent être > 0
- `serie_saison` doit être ≥ 1

Contraintes sur les valeurs que peut prendre `importance` :

```
-- MySQL & PostgreSQL
importance VARCHAR(20) CHECK (importance IN ('Principal', 'Secondaire'))
```

VI. Création des tables

VI.1 Version MySQL

Le script d'initialisation de la Base de données en MySQL est retrouvable dans le dossier .zip sous le nom script_mysql.sql

VI.2 Version PostgreSQL

Le script d'initialisation de la Base de données en PostgreSQL est retrouvable dans le dossier .zip sous le nom script_postgresql.sql

VI.3 Ressources utilisés

- https://starwars.fandom.com/fr/wiki/Accueil
- https://www.starwars.com/databank/
- https://www.imdb.com/

VII. Conclusion

Récapitulatif de la transformation

Notre modèle E/A est composé de :

- **3 entités** (Films, Series, Personnages)
- **2 associations n:m** (Joue_film, Joue_serie)

Il a été transformé en un modèle relationnel composé de :

- **5 tables** au total
- **3 tables d'entités** avec clés primaires
- **2 tables d'associations** avec clés primaires
- **4 contraintes de clés étrangères**