

C O N C E P T I O N
D ' U N E B A S E
D E D O N N É E S

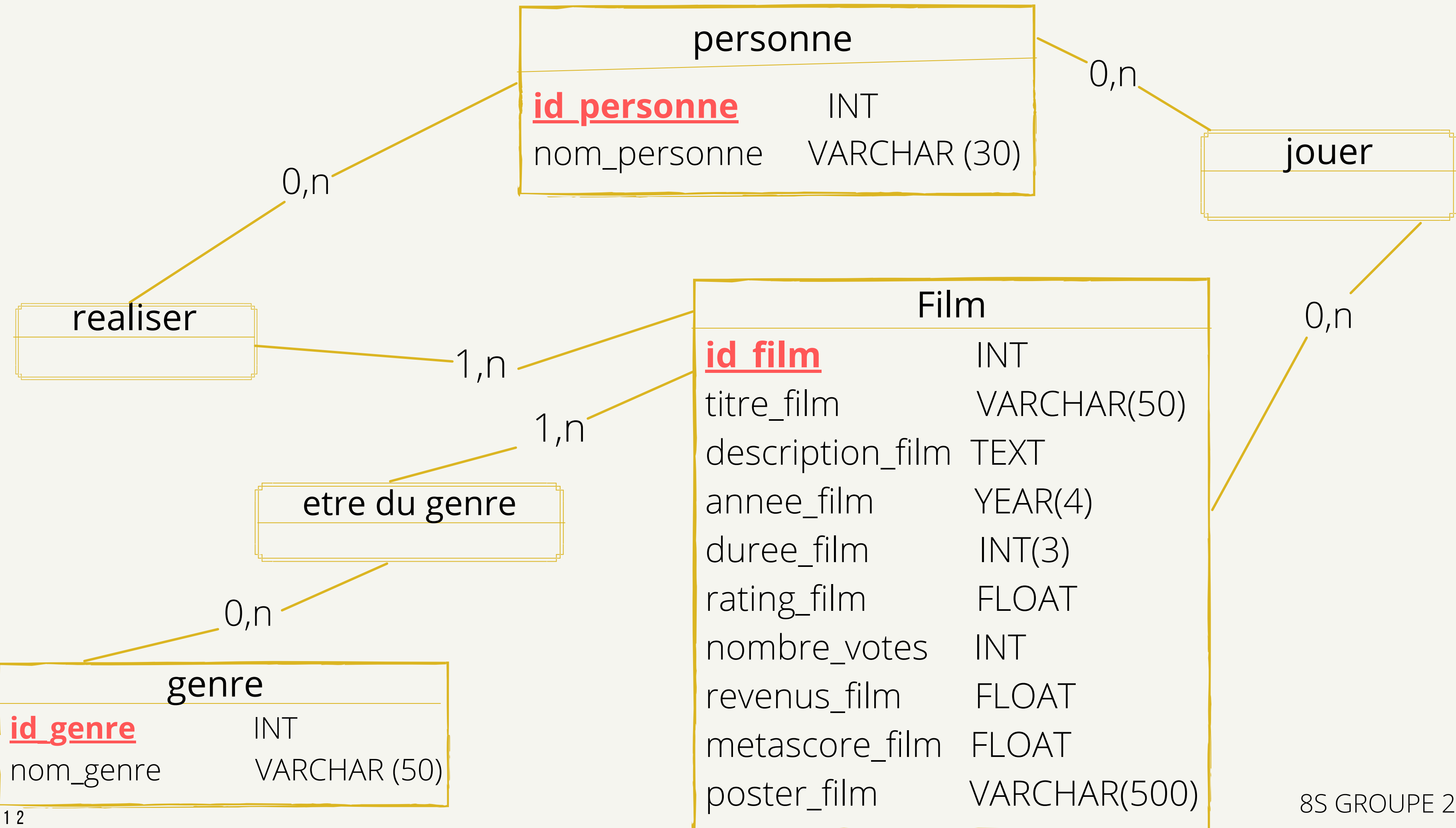
RAPHAËL CANIN, AMANDA NGUYEN, ROMAIN
SAVINO, YANN SOBGUI & RÉMY XIA

Description de notre base de données

2 objectifs:

- Faciliter la recherche précise de films
- Proposer un tri sur des critères flexibles sur un ensemble de films

MODÈLE ENTITÉ- ASSOCIATION



Volumes concernés

- *1 dataset utilisé, issu de IMDB*

<https://www.kaggle.com/omarhanyy/imdb-top-1000>

1000 films auxquels sont rattachés

- 4 acteurs
- 1 réalisateur
- 1 à 3 genres

Méthodes

- Création des tables
- Peuplement de la base par des requêtes générées via Google Sheets
- Création d'un site internet pour présenter le résultat

Résultats : Présentation des requêtes SQL

Peuplement et création

A	B	C	D	E	F	G	H	I
id_film	titre_film	description_film	annee_film	duree_film	rating_film	nombre_votes	revenus_film	metascore_film
1	Guardians of the Galaxy	A group of intergalactic criminals	2014	121	8.1	757074	333.13	76
2	Prometheus	Following clues to the origin of	2012	124	7	485820	126.46	65
3	Split	Three girls are kidnapped by a	2016	117	7.3	157606	138.12	62
4	Sing	In a city of humanoid animals,	2016	108	7.2	60545	270.32	59
5	Suicide Squad	A secret government agency r	2016	123	6.2	393727	325.02	40
6	The Great Wall	European mercenaries search	2016	103	6.1	56036	45.13	42
7	La La Land	A jazz pianist falls for an aspiri	2016	128	8.3	258682	151.06	93
8	Mindhorn	A has-been actor best known for	2016	89	6.4	2490	0	71
9	The Lost City of Z	A true-life drama, centering on	2016	141	7.1	7188	8.01	78
10	Passengers	A spacecraft traveling to a dist	2016	116	7	192177	100.01	41
11	Fantastic Beasts and Where to Find Them	The adventures of writer Newt	2016	133	7.5	333333	333.33	66

```
INSERT INTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`,
`rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`)
VALUES (1,"Guardians of the Galaxy","A group of intergalactic criminals are forced to work
together to stop a fanatical warrior from taking control of the
universe.",2014,121,8.1,757074,333.13,76);
```

#-----
Table: Film
#-----

```
CREATE TABLE film(
  id_film Int NOT NULL ,
  titre_film Varchar (50) NOT NULL ,
  description_film Text NOT NULL ,
  annee_film Year NOT NULL ,
  duree_film Int NOT NULL ,
  rating_film Float NOT NULL ,
  nombre_votes Int NOT NULL ,
  revenus_film Float NOT NULL ,
  metascore_film Float NOT NULL
,CONSTRAINT film_PK PRIMARY KEY (id_film)
)ENGINE=InnoDB;
```

```
fx | ="INSERT INTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES ("& A3 & ",""" & B3 &""",'"
L
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (1,"Guardians of the Galaxy","A group of intergalactic criminals are forced
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (2,"Prometheus","Following clues to the origin of mankind, a team finds a s
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (3,"Split","Three girls are kidnapped by a man with a diagnosed 23 distinct
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (4,"Sing","In a city of humanoid animals, a hustling theater impresario's at
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (5,"Suicide Squad","A secret government agency recruits some of the most
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (6,"The Great Wall","European mercenaries searching for black powder bec
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (7,"La La Land","A jazz pianist falls for an aspiring actress in Los Angeles."
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (8,"Mindhorn","A has-been actor best known for playing the title character
JTO `film`(`id_film`, `titre_film`, `description_film`, `annee_film`, `duree_film`, `rating_film`, `nombre_votes`, `revenus_film`, `metascore_film`) VALUES (9,"The Lost City of Z","A true-life drama, centering on British explorer Col.
```

Difficultés rencontrées

Liée au peuplement via Excel

- Doublons dans le dataset
- Recherche du réalisateur parmi une liste

Plantage de Gaalactic, migration vers IONOS

Requêtes SQL utilisées

- Suggestion en temps réel pour la recherche
- Affichage des différents genres pour la recherche par genre
- Tri des résultats
- Affichage général des résultats
- Recherche par titre de film
- Recherche par réalisateur
- Recherche par acteur(s)
- Recherche par genre(s)

Affichage général des résultats

```
SELECT poster_film, titre_film, annee_film, duree_film, description_film,
rating_film, realisateur.nom_personne AS nom_real,
#Concaténation du nom de tous les acteurs appartenant au même film :
GROUP_CONCAT(DISTINCT acteur.nom_personne SEPARATOR ',') AS acteurs,
#Concaténation du nom de tous les genres appartenant au même film :
GROUP_CONCAT(DISTINCT genre.nom_genre SEPARATOR ', ') AS genres FROM film

#Jointure de table pour accéder aux nom des réalisateurs de chaque film :
# + Alias de table afin de différencier acteurs et réalisateur
JOIN realiser ON film.id_film=realiser.id_film JOIN personne AS realisateur
ON realiser.id_personne=realisateur.id_personne
```



```
#Jointure de table pour accéder aux noms des acteurs de chaque film :
# + Alias de table afin de différencier acteurs et réalisateur
JOIN jouer ON film.id_film=jouer.id_film JOIN personne AS acteur
ON jouer.id_personne=acteur.id_personne

#Jointure de table pour accéder aux noms des genres de chaque film
JOIN etre_du_genre ON film.id_film=etre_du_genre.id_film JOIN genre
ON genre.id_genre=etre_du_genre.id_genre

WHERE ... #spécifique à chaque catégorie

GROUP BY title_film;
```

Recherche par genre(s)

```
# ...  
WHERE film.id_film IN (SELECT film.id_film FROM film  
                        JOIN etre_du_genre ON film.id_film=etre_du_genre.id_film  
                        JOIN genre ON genre.id_genre=etre_du_genre.id_genre  
                        WHERE nom_genre='genre1')  
AND # ... répéter deux fois de plus pour les deux autres genres  
#...
```

Lien vers le site web

<http://prcmfilms.fr/bddr/>