

# Introduction aux bases de données

## Requêtes avec jointures

# Requêtes avec jointures

## Plan de la séance

- Retour sur la dernière séance:
  - Point de vue de l'étudiant
  - Point de vue de l'enseignant.
- Rappels:
  - La commande SELECT
  - La contrainte de FK
- Le produit cartésien
- Jointure interne: INNER JOIN
- Jointure externe: LEFT/RIGHT OUTER JOIN
- Début du laboratoire 3
- Importance des alias. (exemple)

# Rappel, la commande SELECT

## La commande SELECT

- La commande SELECT est la commande la plus simple à utiliser avec SQL. Cette commande n'affecte en rien la base de données et permet d'extraire des données d'une ou plusieurs tables.
- La syntaxe simplifiée n'utilise pas de jointure et elle se présente comme suit :

```
SELECT <nom_de_colonne1,...nom_de_colonnen>  
FROM <nom_de_table>  
[WHERE <condition>]  
[ORDER BY <nom_de_colonne>];
```

# Jointures: Introduction

Pour les explications nous utiliserons les deux tables suivante

Table films

ID_FILM	TITRE	ANNEE	CLASSE	ID_CATEGORIE
1	Pirates des Caraïbes, la Malédiction du Black Pearl	2003	08+	2
2	Pirates des Caraïbes, le Secret du coffre maudit	2006	08+	2
3	Pirates des Caraïbes, Jusqu'au bout du monde	2007	13+	2
4	Élémentaire	2018	gen	5
5	Holmes à la rescousse	2017	gen	5
6	l'incroyable Holmes	2007	gen	5
7	Mission Mars	2019	08+	4
8	Film d'horreur	2006	13+	3
9	Dîner de cons	2000	gen	1
10	Volver	2007	08+	2
11	Le temps va	2020	gen	(null)
12	La covid est finie	2021	gen	(null)
13	La vie est belle	1999	gen	(null)

Table categories

ID_CATEGORIE	NOM_CATEGORIE
1	Comédie
2	Fantastique
3	Horreur
4	Science fiction
5	Action
6	Drame
7	Comédie musicale

# Produit cartésien, Définition

- **Produit cartésien:**

Le produit cartésien est une requête de sélection qui met en jeu plusieurs tables. Pour deux tables, la sélection consiste à afficher la première ligne de la première table avec toutes les lignes de la deuxième table, puis la deuxième ligne de la première table avec toutes les lignes de la deuxième table et ainsi de suite. Ce type de sélection implique beaucoup de redondances.

Exemple :

```
SELECT titre,nom_categorie  
FROM films, categories;
```

Va nous donner comme résultat

91 lignes. (13\*7)

il y a une suite au résultat de la figure)

Il y a **beaucoup de redondance**.

De plus, remarquez que “Le temps va”

A la catégorie Action alors qu’il n’en a pas

TITRE	NOM_CATEGORIE
Volver	Action
Dîner de cons	Action
Film d'horreur	Action
Mission Mars	Action
l'incroyable Holmes	Action
Holmes à la rescousse	Action
Le temps va	Action
Pirates des Caraïbes, Jusqu'au bout du monde	Action
Pirates des Caraïbes, le Secret du coffre maudit	Action
Pirates des Caraïbes, la Malédiction du Black Pearl	Action
La vie est belle	Action
La covid est finie	Action
Élémentaire	Action
Dîner de cons	Comédie
Volver	Comédie

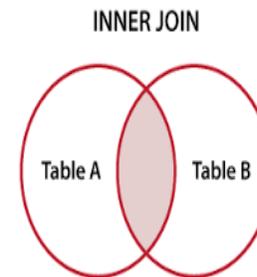
# Jointures: Définitions

- Une jointure

Une jointure est une opération relationnelle qui sert à chercher des lignes (ou des enregistrements) à partir de deux ou plusieurs tables disposant d'un ensemble de valeurs communes, en général les clés primaires.

Il existe plusieurs types de jointure que nous allons aborder au fur et à mesure. Parmi ces jointures, nous avons la jointure interne appelée aussi jointure simple

Une jointure **simple** consiste en un produit cartésien avec un **INNER JOIN** faisant ainsi une restriction sur les lignes. La restriction est faite sur l'égalité de la valeur de deux attributs (cas de deux tables) qui sont la valeur **d'une clé primaire est égale à la valeur d'une clé étrangère.**



Les jointures se font au niveau de la clause **FROM**

# Jointures: Définitions

Syntaxe:

```
SELECT colonne1, colonne2, .....  
FROM [(]nom_table1 INNER JOIN nomTable2  
ON nom_table1.cleEtrangere = nomTable2.clePrimaire []);  
[WHERE <condition>]  
[ORDER BY <nom_de_colonne>];
```

Exemple 1

```
SELECT titre, nom_categorie  
FROM (films INNER JOIN categories  
ON films.id_categorie = categories.id_categorie);
```

Lorsque nous avons INNER JOIN, les parenthèses ne sont pas obligatoires

# Jointures

- Explications

Une requête avec jointure interne **INNER JOIN**, va ramener des résultats des deux tables Films et Categories uniquement s'il y a égalité entre la **clé étrangère code\_categorie de la table films** et la **clé primaire code\_categorie de la table categories**.

Selon le contenu initial des tables, la requête va ramener exactement 10 enregistrements (lignes)

**Les films qui n'ont pas de catégories ne seront pas ramenés. Les catégories qui n'ont pas de films ne seront pas ramenées**

TITRE	NOM_CATEGORIE
Dîner de cons	Comédie
Pirates des Caraïbes, la Malédiction du Black Pearl	Fantastique
Pirates des Caraïbes, le Secret du coffre maudit	Fantastique
Pirates des Caraïbes, Jusqu'au bout du monde	Fantastique
Volver	Fantastique
Film d'horreur	Horreur
Mission Mars	Science fiction
l'incroyable Holmes	Action
Holmes à la rescousse	Action
Élémentaire	Action

# Les jointures

- Lorsqu'un attribut sélectionné est présent dans plus d'une table alors il faut le précéder du nom de la table à partir de laquelle on désire l'extraire. Si le nom de la table n'est pas précisé cela va renvoyer une erreur: « nom de colonne ambiguë »

```
SELECT titre, nom_categorie, categories.id_categorie  
FROM (films INNER JOIN categories  
ON films.id_categorie = categories.id_categorie);
```

- Toutes les tables dont les attributs apparaissent dans la clause **SELECT** ou dans la clause **WHERE** doivent apparaître dans la clause FROM. (dans la jointure)

Dans l'exemple suivant, dans le FROM, nous avons une jointure entre films et categories à cause du WHERE.

```
SELECT titre  
FROM films INNER JOIN categories  
ON films.id_categorie = categories.id_categorie  
WHERE nom_categorie = 'Action';
```

# Les jointures

- Vous pouvez donner un alias aux noms de tables afin **de faciliter la référence** aux tables. Cependant si un alias est donné, alors il faudra utiliser l'alias à la place du nom de la table.
- Un alias est un nom , généralement une lettre, qu'on donne aux tables dans le FROM.

```
SELECT titre, nom_categorie, c.id_categorie  
FROM films f INNER JOIN categories c  
ON f.id_categorie = c.id_categorie;
```

L'exemple suivant va renvoyer une erreur: « **categorie.id\_categorie** » identificateur invalide

```
SELECT titre, nom_categorie, categorie.id_categorie  
FROM films f INNER JOIN categories c  
ON f.id_categorie = c.id_categorie;
```

Donner des Alias n'est pas obligatoire sauf dans un cas que nous verrons plus loin

# Jointures: Autres exemples

Pour la suite utiliserons les deux tables suivante

**Table films**

ID_FILM	TITRE	ANNEE	CLASSE	ID_CATEGORIE
1	Pirates des Caraïbes, la Malédiction du Black Pearl	2003	08+	2
2	Pirates des Caraïbes, le Secret du coffre maudit	2006	08+	2
3	Pirates des Caraïbes, Jusqu'au bout du monde	2007	13+	2
4	Élémentaire	2018	gen	5
5	Holmes à la rescousse	2017	gen	5
6	l'incroyable Holmes	2007	gen	5
7	Mission Mars	2019	08+	4
8	Film d'horreur	2006	13+	3
9	Dîner de cons	2000	gen	1
10	Volver	2007	08+	2
11	Le temps va	2020	gen	(null)
12	La covid est finie	2021	gen	(null)
13	La vie est belle	1999	gen	(null)

**Table acteurs**

ID_ACTEUR	NOM_ACTEUR	PRENOM_ACTEUR	SALAIREMINIMUM	NATIONALITE
1	Depp	Johnny	1000000	Américain
2	Radcliffe	Daniel	2000000	Britannique
3	L'acteur	Georges	800000	Américain
4	Actrice	Alice	1500000	Américaine
5	Saturne	Kévin	1200000	Canadien
6	Lechat	Simba	3000000	Canadien
7	Fafar	Ruby	900000	Canadienne
8	Watson	Emma	1500000	Britannique
9	Monsieur	Spok	2500000	Américain
10	Holmes	Sherlock	1000000	Américain

**Table syActeurs\_film**

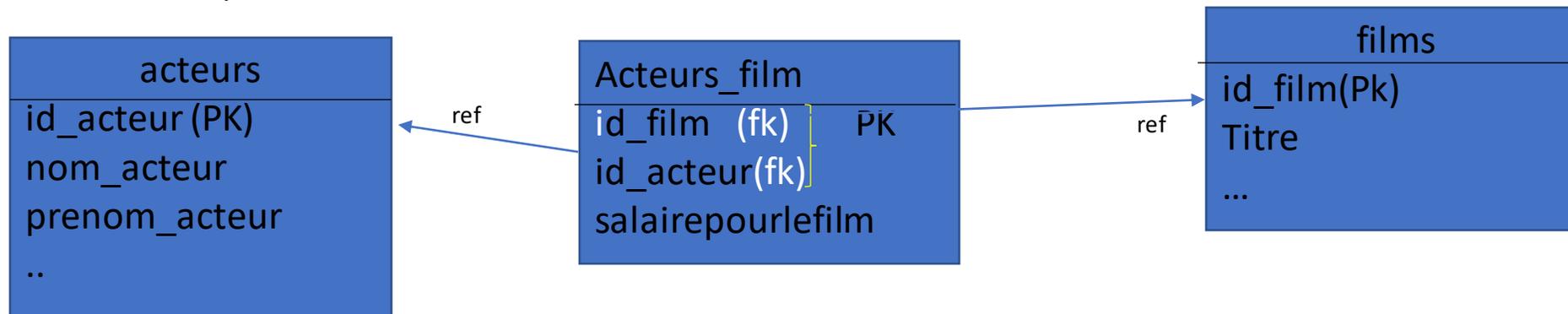
ID_FILM	ID_ACTEUR	SALAIREPOURLEFILM
1	1	1000000
2	1	5000000
3	1	8000000
2	7	1000000
3	5	1500000
1	5	600000

# Jointures, exemples

Nous voulons savoir avoir le nom des acteurs, leurs prénoms ,le titre du film et le salaire qu'ils ont gagné pour le film. La requête serait la suivante:

```
SELECT nom_acteur, prenom_acteur, titre, salairepourlefilm
FROM acteurs
INNER JOIN Acteurs_film ON acteurs.id_acteur = Acteurs_film.id_acteur
INNER JOIN films on films.id_film = Acteurs_film .id_film;
```

Avant de se lancer dans l'écriture de la requête, il est très importants de visualiser le lien entre les 3 tables pour comprendre les jointures. Car n'oublions pas, une jointure c'est juste une valeur de clé étrangère qui est égale à une valeur de clé primaire.



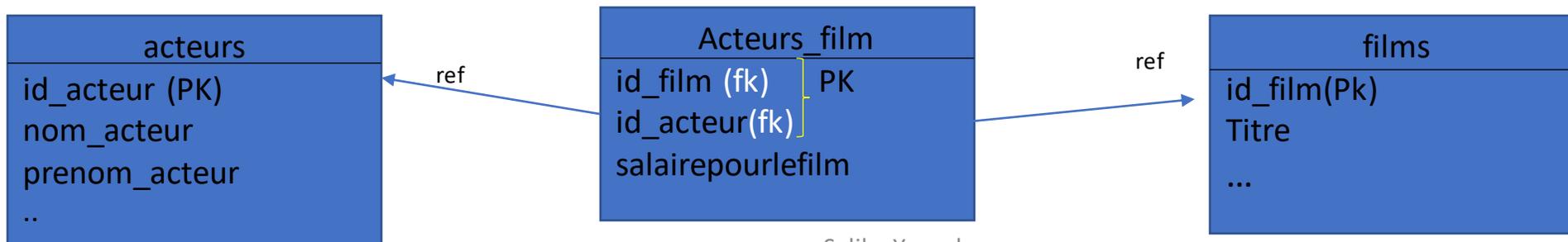
# Jointures, Table de lien

Nous voulons savoir le nom des acteurs, leurs prénoms qui on joué dans le film: Pirates des Caraïbes,Jusqu'au bout du monde

```
SELECT nom_acteur, prenom_acteur
FROM syacteurs
INNER JOIN Acteurs_film ON syacteurs.id_acteur = Acteurs_film .id_acteur
INNER JOIN films ON films.id_film = Acteurs_film.id_film
WHERE titre = 'Pirates des Caraïbes,Jusqu"au bout du monde';
```

Quand on regarde la clause SELECT, nous avons le nom et le prenom de l'acteur donc nous avons besoin de la table **syacteurs**. La clause WHERE indique que nous avons besoin de la table **films**. (à cause du titre du film).

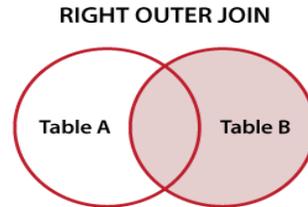
Les deux tables **syacteurs** et **Films** ne sont aps directement liées ensemble. Elles sont liées par la table **syActeurs\_film**. La table syActeurs\_film **est appelée table de lien**.



# Les jointures externes

Définitions :

Jointure externe droite (**RIGHT OUTER JOIN**)



Dans la jointure externe droite, des enregistrements de table à droite de la jointure seront ramenés même si ceux-ci n'ont pas d'occurrences dans l'autre table.

Dans l'exemple suivant, la requête va renvoyer tous les films(titre) ayant une catégorie et toutes les catégories y compris celles qui n'ont pas de films (voir acétate suivante).

La raison est que **categories** est à droite de la jointure (JOIN), donc le système va ramener TOUTES les catégories

```
SELECT titre, nom_categorie
FROM (films RIGHT OUTER JOIN categories
ON films.id_categorie = categories.id_categorie);
```

# Les jointures externes

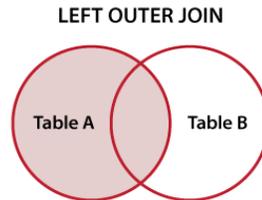
TITRE	NOM_CATEGORIE
Dîner de cons	Comédie
Pirates des Caraïbes, la Malédiction du Black Pearl	Fantastique
Pirates des Caraïbes, le Secret du coffre maudit	Fantastique
Pirates des Caraïbes, Jusqu'au bout du monde	Fantastique
Volver	Fantastique
Film d'horreur	Horreur
Mission Mars	Science fiction
l'incroyable Holmes	Action
Holmes à la rescousse	Action
Élémentaire	Action
(null)	Drame
(null)	Comédie musicale

```
SELECT titre, nom_categorie  
FROM (films RIGHT OUTER JOIN categories  
ON films.id_categorie = categories.id_categorie);
```

# Les jointures externes

Définitions :

Jointure externe Gauche (**LEFT OUTER JOIN**)



Dans la jointure externe gauche, des enregistrements de table à gauche de la jointure seront ramenés même si ceux-ci n'ont pas d'occurrences dans l'autre table.

Dans l'exemple suivant, la requête va renvoyer tous les films(titre) ayant une catégorie et les films qui ne sont dans aucune catégorie. La requête ne va ramener que les catégories ayant des films

La raison est que **films** est à gauche de la jointure (JOIN), donc le système va ramener TOUS les films

```
SELECT titre, nom_categorie
FROM (films LEFT OUTER JOIN Categories
ON Films.id_categorie = Categories.id_categorie);
```

# Les jointures externes

TITRE	NOM_CATEGORIE
Élémentaire	Action
Holmes à la rescousse	Action
l'incroyable Holmes	Action
Dîner de cons	Comédie
Pirates des Caraïbes, la Malédiction du Black Pearl	Fantastique
Pirates des Caraïbes, le Secret du coffre maudit	Fantastique
Pirates des Caraïbes, Jusqu'au bout du monde	Fantastique
Volver	Fantastique
Film d'horreur	Horreur
Mission Mars	Science fiction
Le temps va	(null)
La covid est finie	(null)
La vie est belle	(null)

```
SELECT titre, nom_categorie
FROM (Films LEFT OUTER JOIN Categories
ON films.id_categorie = categories.id_categorie);
```

# Les jointures externes

Si nous avons bien compris, ces deux requêtes donnent le même résultats.

C'est la façon dont vous placez vos tables par rapport à la jointure qui est importante

```
SELECT titre, nom_categorie  
FROM (films LEFT OUTER JOIN categories  
ON films.id_categorie = categories.id_categorie);
```

```
SELECT titre, nom_categorie  
FROM (categories RIGHT OUTER JOIN films  
ON films.id_categorie = categories.id_categorie);
```

# Jointures: importance des Alias

Pour la suite utiliserons les deux tables suivante

**Table films**

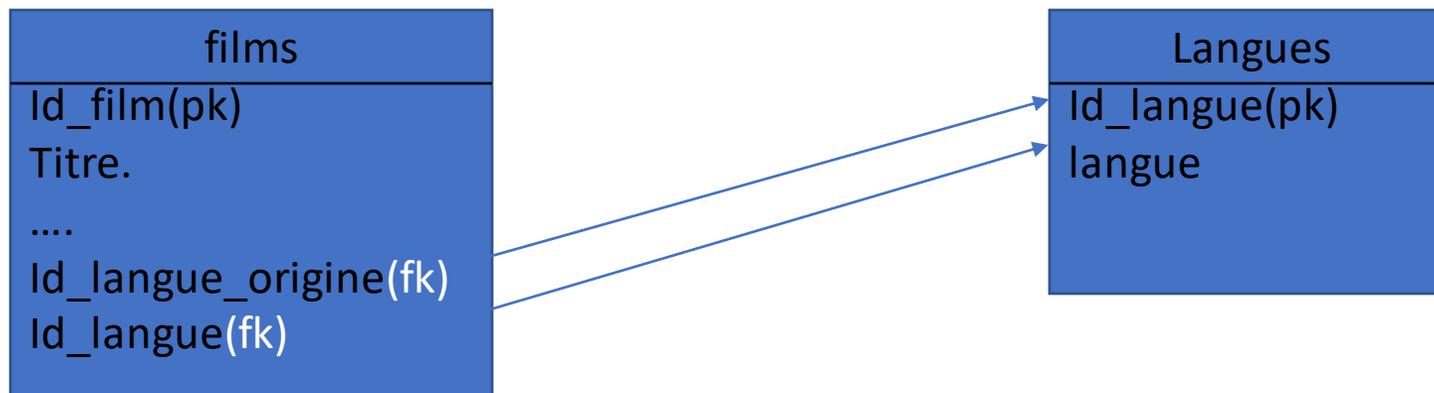
ID_FILM	TITRE	ANNEE	CLASSE	ID_CATEGORIE	ID_LANGUE_ORIGINE	ID_LANGUE
1	Pirates des Caraïbes, la Malédiction du Black Pearl	2003	08+	2	en	fr
2	Pirates des Caraïbes, le Secret du coffre maudit	2006	08+	2	en	fr
3	Pirates des Caraïbes, Jusqu'au bout du monde	2007	13+	2	en	fr
4	Élémentaire	2018	gen	5	en	fr
5	Holmes à la rescousse	2017	gen	5	en	fr
6	l'incroyable Holmes	2007	gen	5	en	fr
7	Mission Mars	2019	08+	4	fr	en
8	Film d'horreur	2006	13+	3	fr	en
9	Dîner de cons	2000	gen	1	fr	es
10	Volver	2007	08+	2	es	de
11	Le temps va	2020	gen	(null)	es	fr
12	La covid est finie	2021	gen	(null)	fr	es
13	La vie est belle	1999	gen	(null)	en	fr

**Table langues**

ID_LANGUE	LANGUE
fr	Français
en	Anglais
es	Espagnol
de	Allemand

# Importance des Alias

Dans la table Films, nous avons `id_langue_dorigine` et `id_langues` qui sont deux FK qui font références à la même table Langues



La question est comment faire pour aller chercher le titre du film, la langue d'origine du film et la langue du film.

On doit faire deux jointures sur la même table langues. La seule façon de la faire est d'utiliser des Alias. C'est comme si la table langues existe deux fois avec des noms différents, Une fois pour les langues d'origine, une autre fois pour la langue du film

# Importance des Alias

Pour la requête suivante, nous avons donné :

l'Alias **o** pour langue d'origine

l'Alias **t** pour langue de traduction

Nous avons donné des alias la colonne langue. Une fois avec Langue\_origine et une autre fois avec Langue\_traduction

Quand on donne un alias aux colonnes, on utilise le mot réservé **AS** ce qui n'est pas le cas pour une table

Le résultat de la requête est sur l'acétate suivante.

```
SELECT titre, o.langue as Langue_origine, t.langue as langue_traduction
FROM films f
INNER JOIN langues o on o.id_langue = f.id_langue_origine
INNER JOIN langues t on t.id_langue = f.id_langue;
```

# Jointures: importance des Alias

◇ TITRE	◇ LANGUE_ORIGINE	◇ LANGUE_TRADUCTION
Élémentaire	Anqlais	Français
l'incroyable Holmes	Anqlais	Français
Holmes à la rescousse	Anqlais	Français
Pirates des Caraïbes, Jusqu'au bout du monde	Anqlais	Français
Pirates des Caraïbes, le Secret du coffre maudit	Anqlais	Français
Pirates des Caraïbes, la Malédiction du Black Pearl	Anqlais	Français
La vie est belle	Anqlais	Français
Le temps va	Espagnol	Français
Mission Mars	Français	Anqlais
Film d'horreur	Français	Anqlais
Dîner de cons	Français	Espagnol
La covid est finie	Français	Espagnol
Volver	Espagnol	Allemand

# Les jointures



Conclusion



Questions