

Internet et Outils
L1/IO2 2007 - 2008
S2-IO2

Bases de données:
Tables Multiples, Schémas

François Armand

`armand@informatique.univ-paris-diderot.fr`

Plan Général Prévisionnel (1/2)

(non contractuel)

• Cours Internet et Outils:

- [1/12] Intro, Internet, Web, XHTML (2H)
- [2/12] XHTML(2H)
- [3/12] CSS (2H)
- [4/12] PHP (2H)
- [5/12] PHP suite (2H)
- [6/12] Introduction MySQL, Table simple (2H)

- [7/12] ► **Partiel (2H) 15 Mars 08 14H-16H** ◀
Amphis 2A et 8C

Plan Général Prévisionnel (2/2)

(non contractuel)

- Bases de données
 - [8/12] **Tables multiples, Schémas (2H)**
 - [9/12] Concepts un peu plus avancés (2H)
 - [10/12] Introduction aux réseaux(2H)
 - [11/12] Réseaux: compléments (2H)
 - [12/12] Révisions (2H)

BD: Plan

- Ce que vous allez découvrir
- Compléments sur SELECT
- Table Simple: Problèmes et Solution
- Modèle Entité / Association
- Du modèle aux tables
- Clés
- Accès à plusieurs tables: jointure
- Ce qu'il faut retenir

Ce que vous allez découvrir

- Pourquoi et comment structurer l'information dans des tables différentes,
- Référencer l'information d'une table depuis une autre,
- Appliquer des contraintes d'intégrité sur les données d'une table,
- Combiner les informations stockées dans des tables différentes... Votre première "jointure"!

BD: Plan

- Ce que vous allez découvrir
- **Compléments sur SELECT**
- Table Simple: Problèmes et Solution
- Modèle Entité / Association
- Du modèle aux tables
- Clés
- Accès à plusieurs tables: jointure
- Ce qu'il faut retenir

SELECT ... IN

- Rechercher des attributs appartenant à un ensemble:

```
mysql> SELECT Titre From FilmSimple  
-> WHERE Nom IN ('Hitchcock',  
-> 'Scott', 'Kurosawa');
```

- Plus simple qu'une suite de OR

```
mysql> SELECT Titre From FilmSimple  
-> WHERE Nom='Hitchcock' OR  
Nom='Scott' OR Nom='Kurosawa';
```

Agréger des résultats

EnCours

Client	Depense
Jules	752
Albert	165
Gaston	321
Albert	1438

```
mysql> SELECT Client, SUM(Depense) FROM
EnCours;
```

```
+-----+-----+
| Client | Depense |
+-----+-----+
| Jules  |    752  |
| Albert |   1603  |
| Gaston |    321  |
| Albert |   1603  |
+-----+-----+
```

Agréger des résultats: GROUP BY

```
mysql> SELECT Client, SUM(Depense) FROM  
EnCours GROUP BY Client;
```

```
+-----+-----+  
| Client | Depense |  
+-----+-----+  
| Jules  |    752  |  
| Albert |   1603  |  
| Gaston |    321  |  
+-----+-----+
```

Agréger des résultats avec condition

```
mysql> SELECT Client, SUM(Depense) FROM
EnCours GROUP BY Client HAVING
SUM(Depense) > 500;
```

Client	Depense
Jules	752
Albert	1603

BD: Plan

- Ce que vous allez découvrir
- Compléments sur SELECT
- **Table Simple: Problèmes et Solution**
- Modèle Entité / Association
- Du modèle aux tables
- Clés
- Accès à plusieurs tables: jointure
- Ce qu'il faut retenir

Rappel: Une liste (de films)

- Pour gérer une liste (de films), on peut utiliser un fichier manipulé avec un éditeur de texte:

```
--> cat films.txt
```

```
Alien 1979 Scott Ridley 1943
```

```
Vertigo 1958 Hitchcock Alfred 1899
```

```
Psychose 1960 Hitchcock Alfred 1899
```

```
Kagemusha 1980 Kurosawa Akira 1910
```

```
Volte-face 1997 Woo John 1946
```

Rappel: Table

Film

Nom de la table

Titre	Année	Nom_Réal	Prénom_Réal	Naissance
Alien	1979	Scott	Ridley	1943
Vertigo	1958	Hitchcock	Alfred	1899
Psychose	1960	Hitchcock	Alfred	1899
Kagemusha	1980	Kurosawa	Akira	1910
Volte-face	1997	Woo	John	1946
Pulp Fiction	1995	Tarantino	Quentin	
Titanic	1997	Cameron	James	1954
Sacrifice	1986	Tarkovski	Andrei	1932

- Informations conformes à une description précise.
- Possibilité de flexibilité
- Attributs définis: Titre, Année, ...

Problèmes avec cette table (1/3)

- Insertion:

- Possibilité d'insérer plusieurs fois le même film... y compris avec des descriptions différentes!
- Question: Qu'est-ce qui distingue 2 films?
- Réponse:... ça dépend!
- Dans la suite on suppose que 2 films différents n'ont pas le même titre... C'est une simplification volontaire et assumée de la réalité!

Problèmes avec cette table (2/3)

- Modifications:
 - Les informations sur les metteurs en scène sont dupliquées pour chaque film qu'ils ont réalisé.
 - Si on a besoin de rectifier une erreur sur l'année de naissance, il faut penser à le faire pour tous les films. Sinon, la table contient des informations incohérentes...
 - D'ailleurs, qu'est-ce qui différencie un réalisateur d'un autre? (question similaire à celle sur les films)

Problèmes avec cette table (3/3)

- Destructurations:

- La suppression d'un film de la table, entraîne la suppression des informations associées sur le réalisateur.
- Si le réalisateur n'était présent que pour un seul film, la suppression de ce film entraîne la disparition des informations relatives au réalisateur.

Solutions

- Représenter les films et les réalisateurs indépendamment les uns des autres
 - Insertions, Mises à jour et destructions indépendantes.
- Identifier les films (et les réalisateurs) pour s'assurer qu'aucun doublon ne figure dans nos tables.
- Lier les films et les réalisateurs sans introduire de redondance d'information.

Solution: ébauche

Titre	Année
Alien	1979
Vertigo	1958
Psychose	1960
Kagemusha	1980
Volte-face	1997
Pulp Fiction	1995
Titanic	1997
Sacrifice	1986

Id.	Nom_Réal	Prénom_Réal	Naissance
1	Scott	Ridley	1943
2	Hitchcock	Alfred	1899
3	Kurosawa	Akira	1910
4	Woo	John	1946
5	Tarantino	Quentin	
6	Cameron	James	1954
7	Tarkovski	Andrei	1932

- Films: 2 films ne peuvent avoir le même titre
- Réalisateurs: 2 réalisateurs peuvent avoir le même nom; on les distingue grâce à un identificateur (Id.)

Solution: ébauche

- Il n'y a plus de redondance dans la base de données:
 - Hitchcock n'apparaît qu'une fois.
 - Plus d'incohérence en cas de mise à jour
- Mais l'information:
 - "Quel est le réalisateur du film?" a disparu.

Solution: ébauche

Titre	Année	Id. Réalisateur
Alien	1979	1
Vertigo	1958	2
Psychose	1960	2
Kagemusha	1980	3
Volte-face	1997	4
Pulp Fiction	1995	5
Titanic	1997	6
Sacrifice	1986	7

Id.	Nom_Réal	Prénom_Réal	Naissance
1	Scott	Ridley	1943
2	Hitchcock	Alfred	1899
3	Kurosawa	Akira	1910
4	Woo	John	1946
5	Tarantino	Quentin	
6	Cameron	James	1954
7	Tarkovski	Andrei	1932

- Ajout d'un attribut dans la table film "Id. Réalisateur"
- Toujours pas de redondance

Solution: Ébauche

- Insertion:
 - Les critères d'unicité d'un film ont été définis, on pourra donc contrôler lors d'une insertion d'un film s'il est déjà présent ou non dans la table.
- Mise à jour:
 - Plus de redondance, donc une mise à jour ne risque pas d'introduire d'incohérence.
- Destruction:
 - La destruction d'un film n'affecte pas le réalisateur.

BD: Plan

- Ce que vous allez découvrir
- Compléments sur SELECT
- Table Simple: Problèmes et Solution
- **Modèle Entité / Association**
- Du modèle aux tables
- Clés
- Accès à plusieurs tables: jointure
- Ce qu'il faut retenir

Modèles E/A

- SQL (et donc MySQL) permet de manipuler plusieurs tables et de reconstruire une information dispersée sur ces différentes tables.
- On s'appuie sur des modèles dits "Entités/Associations" (ou E/A).
- Le modèle permet de définir le schéma de la base.
- Modèles E/A (1976) à la base de méthodes de conception comme OMT (UML) ou Merise.

Définir les Entités

- Entité:
 - Objet ou concept identifiable et pertinent pour l'application
 - Être capable de distinguer une entité d'une autre
 - Chaque entité a donc une identité propre
 - Identité nommée: identifiant ou clé
 - Exemple: Le film "Shrek", l'acteur "Michel Simon"
 - Trouver les entités utiles pour l'application
 - Propriétés de l'entité (nom, date de naissance,...)
 - Propriétés ou attributs

Définir les Entités

- Pour définir les entités, on ne s'intéresse pas aux cas particuliers (Hitchcock) mais aux groupes (les réalisateurs).
- Si on veut aussi décrire les acteurs des films, faut-il définir une entité "acteur" et une entité "réalisateur", ou une entité "artiste"?
- Représentation des entités:

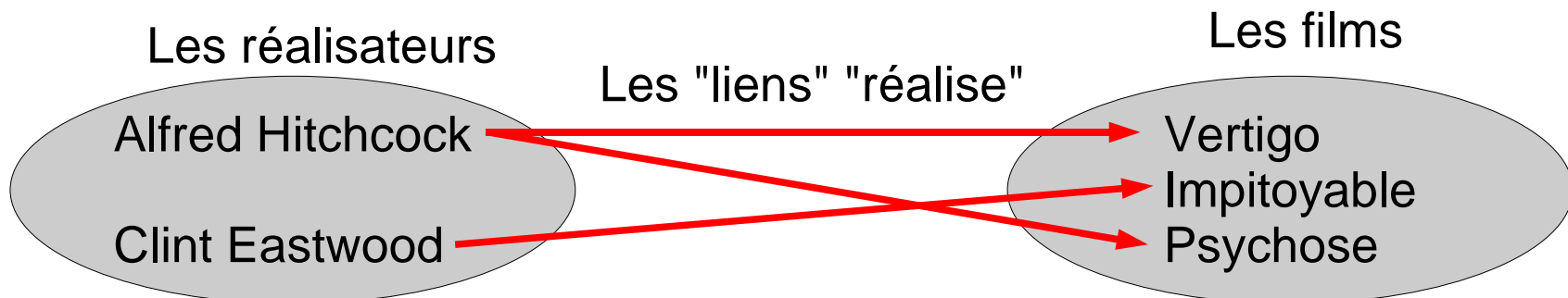
Artiste
Identificateur
Nom
Prénom
Année Naissance

Film
Titre
Année réalisation
Genre
Résumé

Définir les Associations

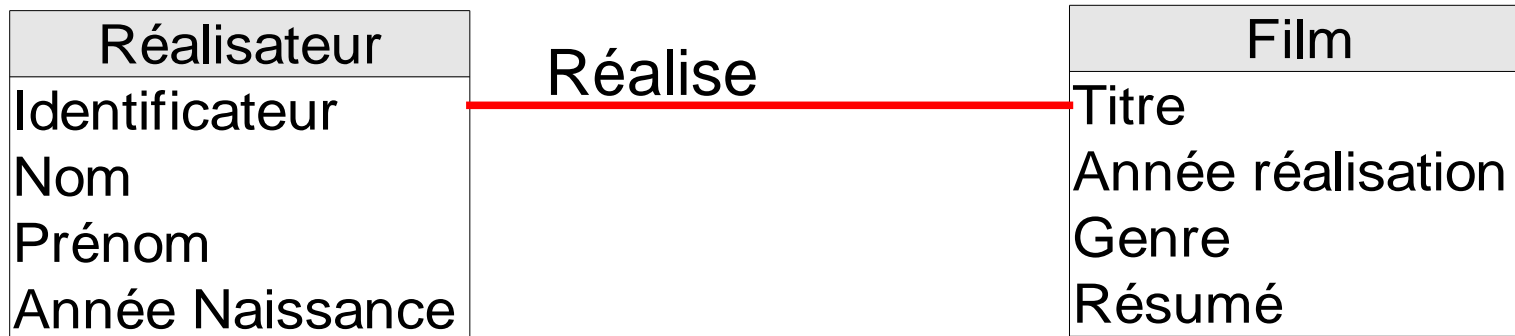
• Association

- Définit une relation entre entités
- Utile pour l'application
- Exemples:
 - Un réalisateur "réalise" un film; ou réciproquement un film "est réalisé" par un réalisateur.
 - Un acteur "joue" dans un film.



Définir les Associations

- Association:
 - représentation d'une association complétant la représentation des entités:

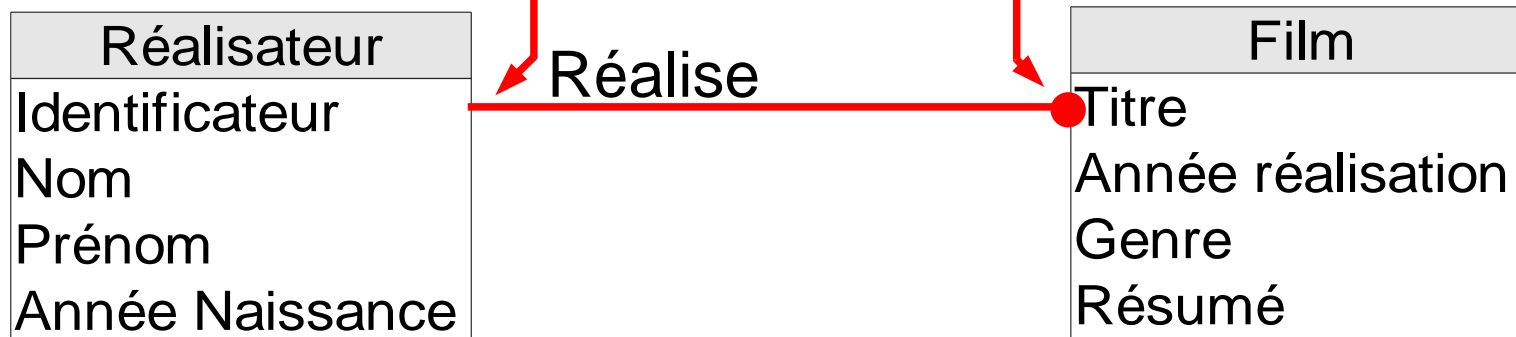


Définir les Associations

- Un réalisateur peut réaliser plusieurs films
- Un film ne peut-être réalisé que par un réalisateur...
 - Dans la réalité, c'est évidemment faux... Ici on a décidé de ne pas prendre en compte ce type de situation.
- Une association a une cardinalité

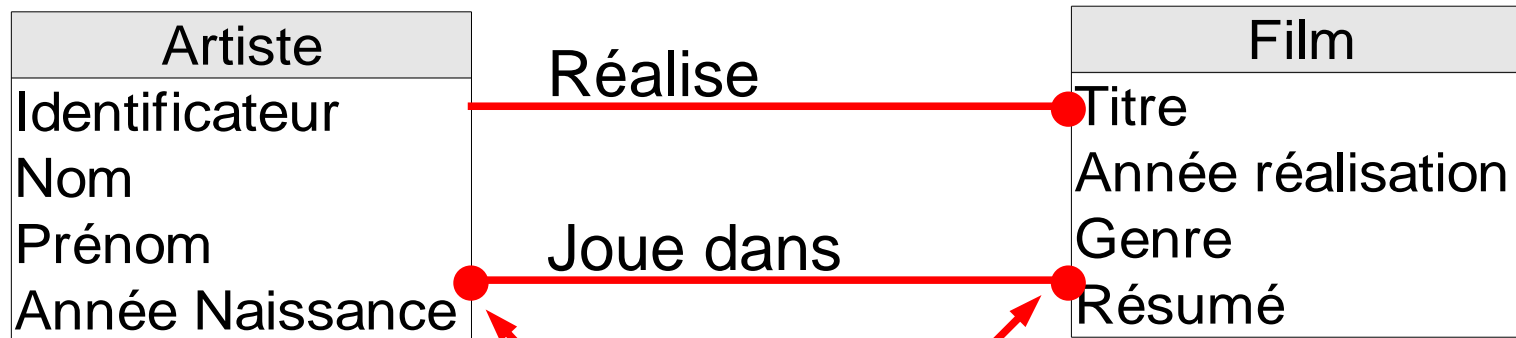
Cardinalité d'une Association

- Le ● en terminaison de la liaison indique que la relation concerne plusieurs objets de ce type:
 - Un réalisateur peut réaliser plusieurs films
 - Un film n'est réalisé que par un seul réalisateur



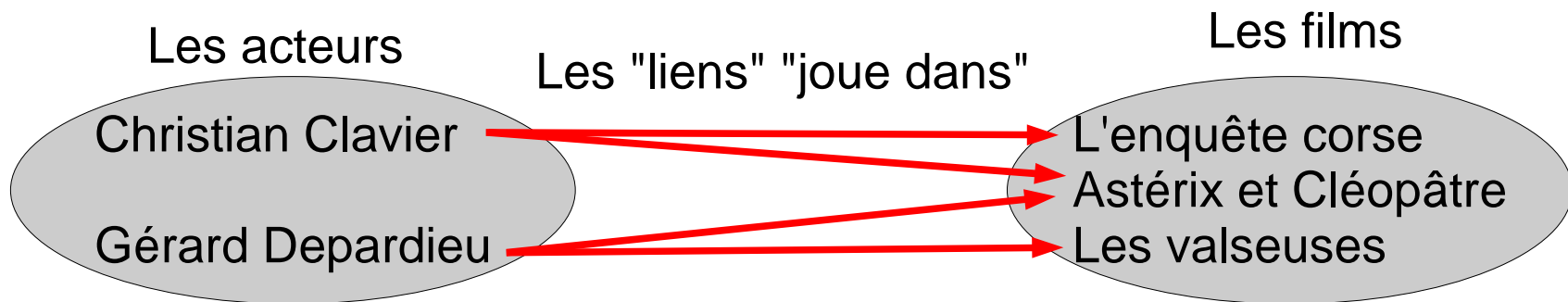
Cardinalité d'une Association

- Le ● en terminaison de la liaison indique que la relation concerne plusieurs objets de ce type:
 - Un artiste peut jouer dans plusieurs films,
 - Plusieurs artistes jouent (peuvent jouer) dans un film



Associations

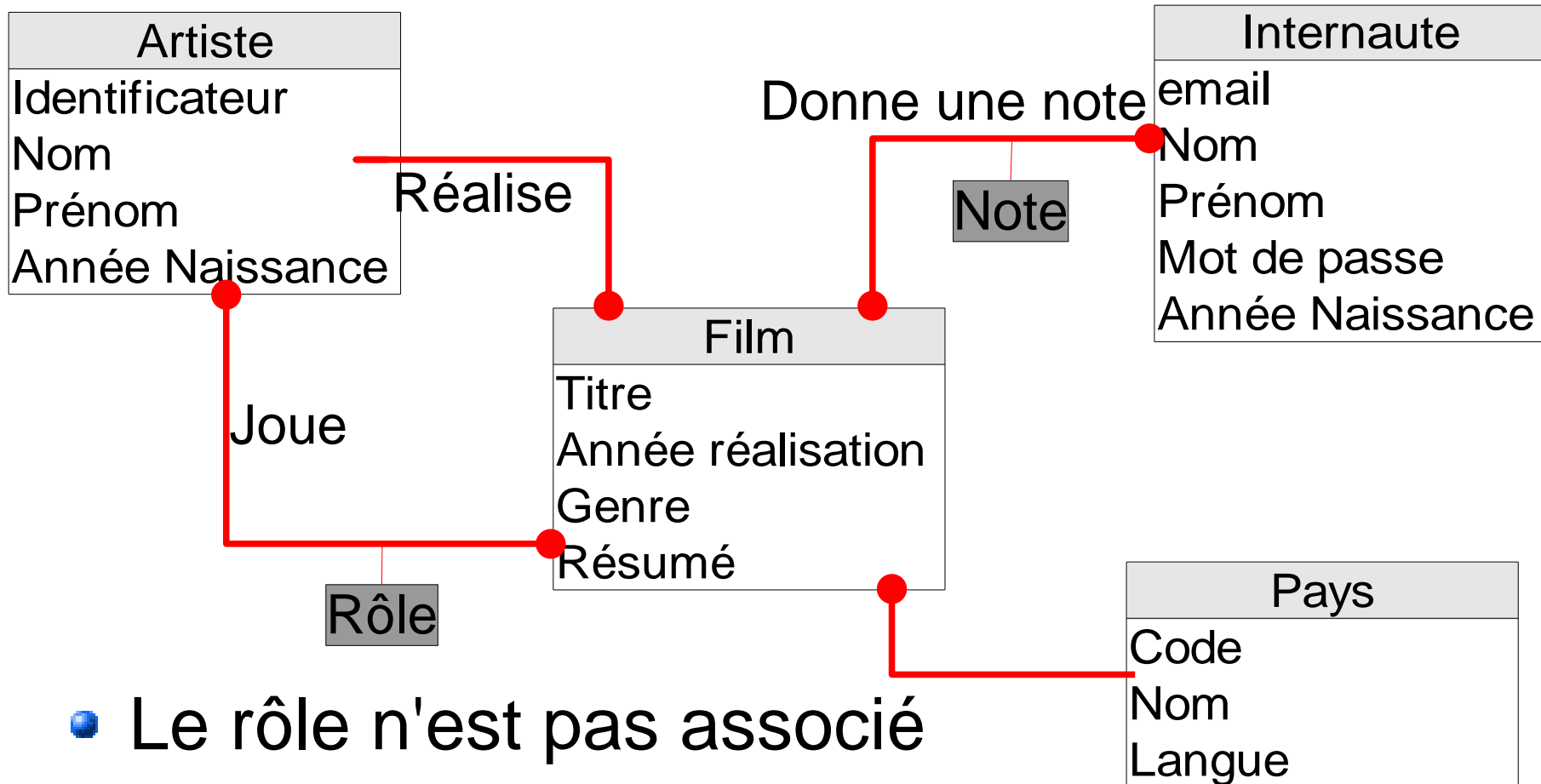
- L'association "joue" décrite ci-avant correspond au modèle présenté ci-dessous.



Une base plus complexe

- On veut représenter:
 - Des films,
 - Les réalisateurs et les acteurs qui jouent,
 - Les pays où ces films ont été réalisés,
- Permettre à des "internauts" :
 - de noter les films enregistrés dans cette base.

Schéma "complexe"



- Le rôle n'est pas associé
 - au film ou à l'acteur, mais à la participation de l'acteur dans un film.

BD: Plan

- Ce que vous allez découvrir
- Compléments sur SELECT
- Table Simple: Problèmes et Solution
- Modèle Entité / Association
- **Du modèle aux tables**
- Clés
- Accès à plusieurs tables: jointure
- Ce qu'il faut retenir


Transcription d'un modèle

- Dans le modèle E/A on dispose de:
 - entités
 - associations
- Dans le modèle SQL, on ne dispose que de :
 - tables!
- Les associations seront représentées par des références d'une table vers une autre.
- Ces références se font en utilisant les "clés" des tables référencées.


Tables et Clés

- Une **clé d'une table**:
 - Plus petit sous-ensemble d'attributs permettant d'identifier une ligne de manière unique et non ambiguë
- Une table a toujours une clé dite
 - **clé principale ou clé primaire**
- Film: Titre, Année, Genre, Résumé
- Artiste: Identificateur, Nom, Prénom, Naissance
- Internaute: e-mail, Nom, Prénom, Naiss., mdp.
- Pays: Code, Nom, Langue

Représentation des associations

- De un (A) à plusieurs (B) 
- Définir les tables correspondant aux entités A et B
- La Clé de A devient un attribut de B.
- Exemple:
 - Film:
 - Titre, Année, Genre, Résumé + *id_réal* + *code_pays*
- La clé ID_réalisateur est
 - Clé **principale** de la table "Artiste"
 - Clé **étrangère** de la table Film.
 - Le nom de la clé peut différer. (Identificateur vs id_réal)

Représentation des associations

- De plusieurs (A) à plusieurs (B) 
- Définir les tables correspondant aux entités A et B
- Définir une table représentant l'association
- La clé primaire de cette table est la paire :
 - clé de A + clé de B (2 clés étrangères)
- Si nécessaire, on peut ajouter des attributs à cette table. Ex: Table "Artiste Joue dans":
 - Clé primaire: Identificateur artiste + Titre du film
 - Attributs: Nom du rôle joué.
 - On suppose qu'un artiste ne joue qu'un rôle dans un film!

Exemples

Films

Titre	Année	Id. Réalisateur	Pays
Alien	1979	1	USA
Vertigo	1958	2	USA
Kagemusha	1980	3	JPN

Rôles

Titre	Id. Artiste	Rôle
Impitoyable	1	William Munny
Impitoyable	5	Little Bill
Ennemi d'état	7	Robert Dean

Artistes

Id.	Nom_Réal	Prénom_Réal	Naissance
1	Scott	Ridley	1943
2	Hitchcock	Alfred	1899
3	Kurosawa	Akira	1910
4	Eastwood	Clint	1930
5	Hackman	Gene	1930
6	Scott	Tony	1930
7	Smith	Will	1968

- La table Rôles a sa clé primaire composée de deux clés étrangères (Films et Artistes) et un attribut spécifique de cette association

MySQL: Création des tables

```
CREATE TABLE Internaute (  
  email VARCHAR(50) NOT NULL,  
  Nom VARCHAR(20) NOT NULL,  
  Prenom VARCHAR(20),  
  Mot_De_Passe VARCHAR(60) NOT NULL,  
  Naissance INTEGER);
```

NOT NULL: impose que l'attribut contienne une valeur.

MySQL: Création des tables

CREATE TABLE Notation (

Titre **VARCHAR(50) NOT NULL,**

email **VARCHAR(50) NOT NULL,**

Note **INTEGER DEFAULT 0**);

DEFAULT: Si la valeur de l'attribut n'est pas spécifiée lors d'une insertion sa valeur sera celle définie par la clause **DEFAULT**.

BD: Plan

- Ce que vous allez découvrir
- Compléments sur SELECT
- Table Simple: Problèmes et Solution
- Modèle Entité / Association
- Du modèle aux tables
- **Clés**
- Accès à plusieurs tables: jointure
- Ce qu'il faut retenir

Contraintes d'intégrité

- **Contrainte d'intégrité** : règle que l'on demande au système de garantir
 - Un attribut doit toujours avoir une valeur (NOT NULL)
 - Un (ensemble d') attribut(s) constitue(nt) la clé de la table
 - Un attribut d'une table est lié à la clé primaire d'une autre table
 - La valeur d'un attribut est unique au sein d'une table
 - Un attribut ne peut prendre qu'une des valeurs prédéfinies d'un ensemble.

Clés Primaires

```
CREATE TABLE Internaute (  
    email VARCHAR(50) NOT NULL,  
    Nom VARCHAR(20) NOT NULL,  
    Prenom VARCHAR(20),  
    Mot_De_Passe VARCHAR(60) NOT NULL,  
    Naissance INTEGER,  
    PRIMARY KEY(email));
```

- Toute table devrait avoir une clé primaire
- Difficile de changer de clé primaire à postériori.

Clés Primaires à plusieurs attributs

CREATE TABLE Notation (

Titre **VARCHAR(50) NOT NULL**,

email **VARCHAR(50) NOT NULL**,

Note **INTEGER DEFAULT 0**,

PRIMARY KEY (Titre, email));

- Une clé primaire peut-être définie par plusieurs attributs.
- Les attributs utilisés doivent être "NOT NULL".

Clés Secondaires

- ```
CREATE TABLE Artiste (
 Ident INTEGER NOT NULL AUTO INCREMENT ,
 Nom VARCHAR(50) NOT NULL,
 Prenom VARCHAR(50) NOT NULL,
 Naissance INTEGER,
 PRIMARY KEY (Ident),
 UNIQUE (Nom, Prenom));
```
- Nom, Prenom: clé secondaire
  - MySQL offre **AUTO\_INCREMENT**, pas SQL...

# Exemple

```
mysql> INSERT INTO Artiste (Nom, Prenom,
-> Naissance) VALUES ('Scott', 'Ridley', 1943);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> INSERT INTO Artiste (Nom, Prenom,
-> Naissance) VALUES ('Hitchcock', 'Alfred',
-> 1899);
Query OK, 1 row affected (0.00 sec)
```

- Pas besoin de préciser la clé "Ident".

# Exemples

```
mysql> SELECT * FROM Artiste;
```

```
+-----+-----+-----+-----+
| ident | Nom | Prenom | Naissance |
+-----+-----+-----+-----+
| 1 | Scott | Ridley | 1943 |
| 2 | Hitchcock | Alfred | 1899 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql>
```

- La clé a été correctement générée

# Exemple: Clé fournie par utilisateur

```
mysql> INSERT INTO Artiste (Ident, Nom, Prenom,
-> Naissance) VALUES (14, 'Woo', 'John', 1946);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM Artiste;
```

| ident | Nom       | Prenom | Naissance |
|-------|-----------|--------|-----------|
| 1     | Scott     | Ridley | 1943      |
| 2     | Hitchcock | Alfred | 1899      |
| 14    | Woo       | John   | 1946      |

```
3 rows in set (0.00 sec)
```

# Exemple: reprise incrémentation

```
mysql> INSERT INTO Artiste (Nom, Prenom,
-> Naissance) VALUES ('Kurosawa', 'Akira',
-> 1910);
```

Query OK, 1 row affected (0.00 sec)

```
mysql> SELECT * FROM Artiste;
```

| ident | Nom       | Prenom | Naissance |
|-------|-----------|--------|-----------|
| 1     | Scott     | Ridley | 1943      |
| 2     | Hitchcock | Alfred | 1899      |
| 14    | Woo       | John   | 1946      |
| 15    | Kurosawa  | Akira  | 1910      |

# Exemple: erreur

- Tentative insertion clé existante:

```
mysql> INSERT INTO Artiste (Ident, Nom, Prenom,
-> Naissance) VALUES (2, 'Woo', 'John',
-> 1946);
```

```
ERROR 1062 (23000) : Duplicata du champ '2'
pour la clef 1
```

```
mysql>
```

# Exemple: Erreur

- Tentative insertion duplication clé secondaire:

```
mysql> INSERT INTO Artiste (Nom, Prenom,
-> Naissance) VALUES ('Woo', 'John', 1946);
```

```
ERROR 1062 (23000): Duplicata du champ
'Woo-John' pour la clef 2
```

```
mysql>
```

# Clés Étrangères

```
CREATE TABLE Films (
 Titre VARCHAR(50) NOT NULL,
 Annee INTEGER NOT NULL,
 Id_Real INTEGER,
 Code_Pays INTEGER,
 PRIMARY KEY (Titre),
 FOREIGN KEY (Id_Real) REFERENCES Artiste,
 FOREIGN KEY (Code_Pays) REFERENCES Pays);
```

# Clés Étrangères

- Lors d'une insertion dans la table Films,
- Lors de toute modification affectant le lien entre les tables
- Vérifications :
  - Valeur de Id\_real est une valeur correcte lors d'une insertion dans Films,
  - Destruction "Artiste" non référencé depuis Films,
  - Modification de Ident dans Artiste ou de Id\_Real dans Films.
- Pas supporté par MySQL... (sauf tables InnoDB)

# Énumérations

- Clause CHECK de SQL
- Mêmes expressions que la clause WHERE des requêtes SQL
- Dans la table Films:
  - Annee **INTEGER**  
**CHECK** (Annee **BETWEEN** 1980 **AND** 2000) **NOT NULL**);
  - genre **VARCHAR**(12)  
**CHECK** (genre **IN** ('Documentaire', 'Western', 'Drame')));
- CHECK accepté mais pas traité par MySQL

# Enumérations

```
CREATE TABLE Films (
 Titre VARCHAR(50) NOT NULL,
 Annee INTEGER
 CHECK (Annee BETWEEN 1980 AND 2000) NOT NULL,
 Genre VARCHAR(20)
 CHECK (genre IN ('Documentaire', 'Western', 'Drame')),
 Id_Real INTEGER,
 Code_Pays INTEGER,
 PRIMARY KEY (Titre),
 FOREIGN KEY (Id_Real) REFERENCES Artiste,
 FOREIGN KEY (Code_Pays) REFERENCES Pays);
```

# BD: Plan

- Ce que vous allez découvrir
- Compléments sur SELECT
- Table Simple: Problèmes et Solution
- Modèle Entité / Association
- Du modèle aux tables
- Clés
- **Accès à plusieurs tables: jointure**
- Ce qu'il faut retenir

# Accéder à plusieurs tables

- Recomposer l'information dont on disposait avec une seule table: Titre, Réalisateur.

```
mysql> SELECT * FROM Artiste;
```

| ident | Nom       | Prenom | Naissance |
|-------|-----------|--------|-----------|
| 1     | Scott     | Ridley | 1943      |
| 2     | Hitchcock | Alfred | 1899      |
| 3     | Woo       | John   | 1946      |
| 4     | Kurosawa  | Akira  | 1910      |
| 5     | Tarkovski | Andrei | 1932      |
| 6     | Cameron   | James  | 1954      |

# Accéder à plusieurs tables

```
mysql> SELECT Titre, Annee, Id_Real FROM
 Films;
```

```
+-----+-----+-----+
| Titre | Annee | Id_Real |
+-----+-----+-----+
| Alien | 1979 | 1 |
| Sacrifice | 1986 | 5 |
+-----+-----+-----+
```

```
2 rows in set (0.00 sec)
```

- **SELECT** permet de référencer des champs de différentes tables.... Essai:
- **SELECT** Titre, Nom **FROM** Films, Artiste;

# Accéder à plusieurs tables

```
mysql> SELECT Titre, Nom FROM Films, Artiste;
```

```
+-----+-----+
| Titre | Nom |
+-----+-----+
Alien	Scott
Sacrifice	Scott
Alien	Hitchcock
Sacrifice	Hitchcock
Alien	Woo
Sacrifice	Woo
Alien	Kurosawa
Sacrifice	Kurosawa
Alien	Tarkovski
Sacrifice	Tarkovski
Alien	Cameron
Sacrifice	Cameron
+-----+-----+
```

Ne donne pas le résultat  
escompté!

Pas de lien explicité entre les  
deux tables.

# Accéder à plusieurs tables

```
mysql> SELECT Titre, Nom FROM Films, Artiste
 -> WHERE Films.Id_Real=Artiste.Ident;
```

```
+-----+-----+
| Titre | Nom |
+-----+-----+
| Alien | Scott |
| Sacrifice | Tarkovski|
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql>
```

- Accès à des informations contenues dans plusieurs tables: JOINTURE.

# BD: Plan

- Ce que vous allez découvrir
- Compléments sur SELECT
- Table Simple: Problèmes et Solution
- Modèle Entité / Association
- Du modèle aux tables
- Clés
- Accès à plusieurs tables: jointure
- **Ce qu'il faut retenir**

# Ce qu'il faut retenir

- Etablir un modèle entité/association des informations
- Convertir en table(s)
- Déterminer et exprimer les contraintes d'intégrité
  - Clés primaires,
  - Clés secondaires,
  - Clés étrangères
- Reconstruire une information complexe
  - Jointure