



La méthode Merise

TP du vidéo club

Créer une base de données depuis un texte brut.

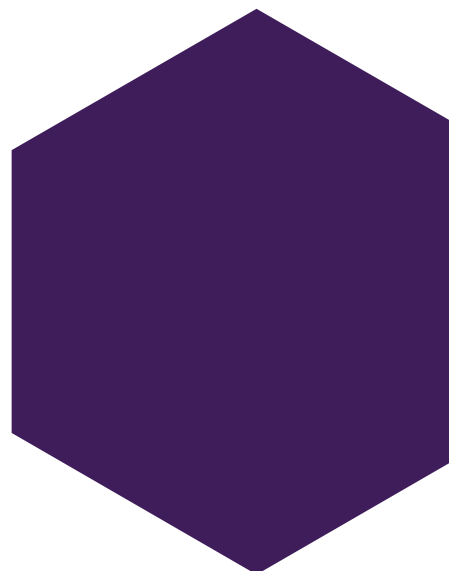
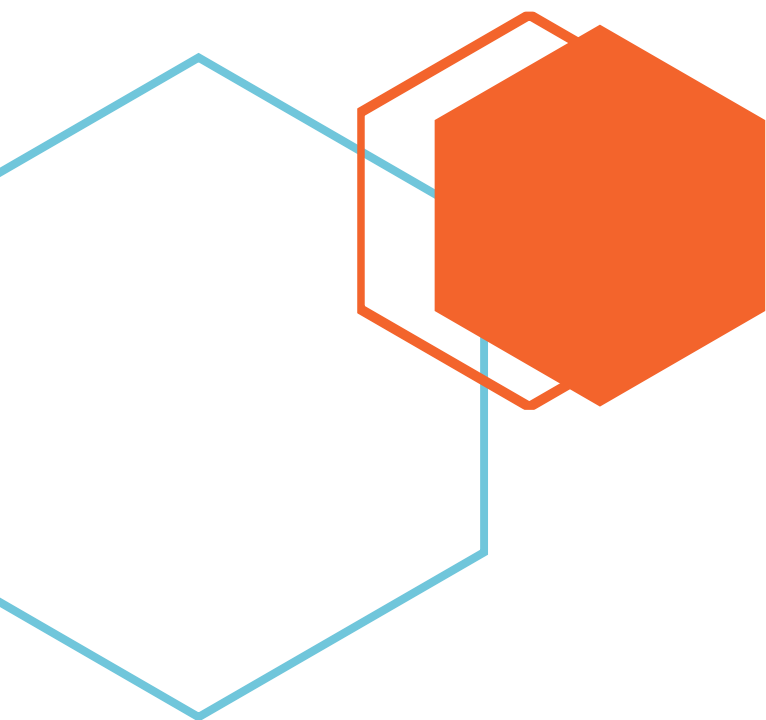




Table des matières

Mise en valeur des données	2
Premier passage :	2
Second passage :	2
Analyse : L'interview	3
Règles de gestion.....	4
Dictionnaire de données.....	5
Dépendances fonctionnelles simples	6
Dépendances fonctionnelles composées.....	7
Modèle conceptuel de données	8
Modèle logique de données	9
Crédits	10

Mise en valeur des données

Premier passage :

Dix magasins de location de cassettes vidéo se sont regroupés pour mettre en commun les cassettes dont ils disposent et ont fondé un club de location. À la suite d'une rencontre avec les représentants de ce club, il ressort que chaque point de vente disposera d'un terminal clavier-écran relié à un site central et qu'il faudra pouvoir prendre en compte les éléments suivants :

- Un **client** qui s'inscrit au club verse une **caution**.
- Suivant le montant de cette caution il aura le droit d'emprunter en même temps de 1 à 6 cassettes.
- Les **cassettes** empruntées doivent être retournées sous 3 jours dans n'importe quelle **boutique** du club.
- Plusieurs cassettes peuvent contenir le même **film**.
- Un film est rattaché à un **genre** cinématographique (**nom** et **type de public**) et est caractérisé par sa **durée**, son **réalisateur** et la liste des **acteurs** principaux.
- Une location n'est permise que si le client est en règle (pas de dépassement du nombre d'emprunts maximum, pas de cassette en retard).
- La consultation d'un client permettra d'obtenir son **nom**, son **adresse**, son nombre d'emprunts en cours, la liste des numéros de cassettes et des titres qu'il a actuellement empruntés.
- La consultation d'un genre permettra d'obtenir la liste des films de ce genre disponibles dans le magasin.
- Périodiquement, on veut obtenir la liste des retardataires ; on veut pour chaque cassette non retournée à temps les informations suivantes :
 - Nom, adresse du client, **date de l'emprunt**, **numéro(s)** de cassette et **titre** du (des) film(s) concerné(s).
- On veut pouvoir connaître pour chaque cassette (identifiée par une numérotation commune aux dix **magasins**) où elle est, quand elle a été **mise en service**, quel film y est enregistré, combien de fois elle a déjà été louée, et quel est son **état** (de très bon à mauvais).

Second passage :

- Un **client** qui s'inscrit au club verse une **caution**.
- Les **cassettes** empruntées doivent être retournées dans sous 3 jours dans n'importe quelle **boutique** du club.
- Plusieurs cassettes peuvent contenir le même **film**.
- Un film est rattaché à un **genre** cinématographique (**nom** et **type** de public) et est caractérisé par sa **durée**, son **réalisateur** et la liste des **acteurs** principaux.
- **Nom**, **adresse** du client, **date de l'emprunt**, **numéro(s)** de cassette et **titre** du (des) film(s) concerné(s).
- On veut pouvoir connaître pour chaque cassette (identifiée par une numérotation commune aux dix magasins) où elle est, **quand elle a été mise en service**, quel film y est enregistré, combien de fois elle a déjà été louée, et quel est son **état** (de très bon à mauvais).



Analyse : L'interview

Est-ce qu'un film peut avoir plusieurs genres ?

Non



Règles de gestion

Un acteur joue dans 1 ou plusieurs films (1,n)

Un film peut être joué par 0 ou plusieurs acteurs (0,n ex : Microcosmos)

Un film est rattaché à un et un seul type (1,1)

Un type est rattaché à un ou plusieurs films (1,n)

Un film peut être enregistré sur 0 ou plusieurs cassettes (0,n)

Une cassette a enregistré un et un seul film (1,1)

Une cassette peut être empruntée par 0 ou plusieurs clients (0,n)

Un client peut emprunter 0 a plusieurs cassettes (0,n)

Une cassette peut être contenue dans 0 ou plusieurs magasins (0,n)

Un magasin peut contenir 0 a plusieurs cassettes (0,n)

Un client fréquente un ou plusieurs magasins (1,n)

Un magasin peut être fréquenté par 0 ou plusieurs clients (0,n)

Un client est enregistré à une et une seule adresse (1,1)

Une adresse peut enregistrer 0 a plusieurs clients (0,n)

Une adresse peut localiser 0 ou magasin (0,1)

Un magasin est localisé à 1 et une seule adresse (1,1)

Dictionnaire de données

Mnémonique	Description	Type	Length	Contraintes	
client_id	Identifiant client	N	10	Identifiant	
client_name	Nom du client	A	50	Obligatoire	
client_firstname	Prénom du client	A	50	Obligatoire	
client_enrollment	Date d'inscription du client	DATETIME	-	Obligatoire	
client_deposit	Montant de la caution	N	3	Obligatoire	
type_id	Identifiant du type	N	10	Identifiant	
type_name	Nom du type	A	50	Obligatoire	
type_public	Type du public concerné	AN	50	Obligatoire	
movie_title	Titre du film	AN	50	Identifiant	
movie_director	Réalisateur du film	A	100	Identifiant	
movie_duration	Durée du film	N	5	Obligatoire	(en minutes, ex : 124 min)
actor_name	Nom de l'acteur	A	50	Identifiant	
actor_firstname	Prénom de l'acteur	A	50	Identifiant	
actor_birth	Date de naissance de l'acteur	DATE	-	Identifiant	
actor_sex	Sexe de l'acteur	A	20	Facultatif	
tape_id	Numéro de la cassette	N	10	Identifiant	
tape_registration	Date de mise en service	DATETIME	-	Obligatoire	
tape_state	Etat de la cassette	AN	50	Obligatoire	
borrow_start	Date de l'emprunt	DATETIME	-	Obligatoire	>= tape_registration
borrow_end	Date de retour de la cassette par le client	DATETIME	-	Facultatif	>= loan_start
shop_name	Nom de la boutique	AN	50	Identifiant	
address_id	Identifiant de l'adresse	N	10	Identifiant	
address_zipcode	Code postal de l'adresse	AN	5	Obligatoire	
address_city	Ville de l'adresse	A	50	Obligatoire	
address_number	Numéro de la rue	AN	10	Facultatif	
address_street_name	Nom de la rue	AN	100	Obligatoire	

Dépendances fonctionnelles simples

	client_id	type_id	movie_title	movie_director	actor_name	actor_firstname	actor_birth	tape_id	shop_name	address_id
client_name	1									
client_firstname	1									
client_enrollment	1									
client_deposit	1									
type_id			1							
type_name		1								
type_public		1								
movie_title			1							
movie_director			1							
movie_duration			1							
actor_name					1					
actor_firstname					1					
actor_birth					1					
actor_sex					1					
tape_registration								1		
tape_state								1		
shop_name										1
address_id	1								1	
address_zipcode										1
address_city										1
address_number										1
address_street_name										1

client_id → client_name , client_firstname , client_enrollment , client_deposit , address_id

type_id → type_name , type_public ,

(movie_title, movie_director) → type_id , movie_title , movie_director , movie_duration

(actor_name, actor_firstname, actor_birth) → actor_name , actor_firstname , actor_birth , actor_sex

tape_id → tape_registration , tape_state ,

shop_name → address_id ,

address_id → shop_name , address_zipcode , address_city , address_number , address_street_name



Dépendances fonctionnelles composées

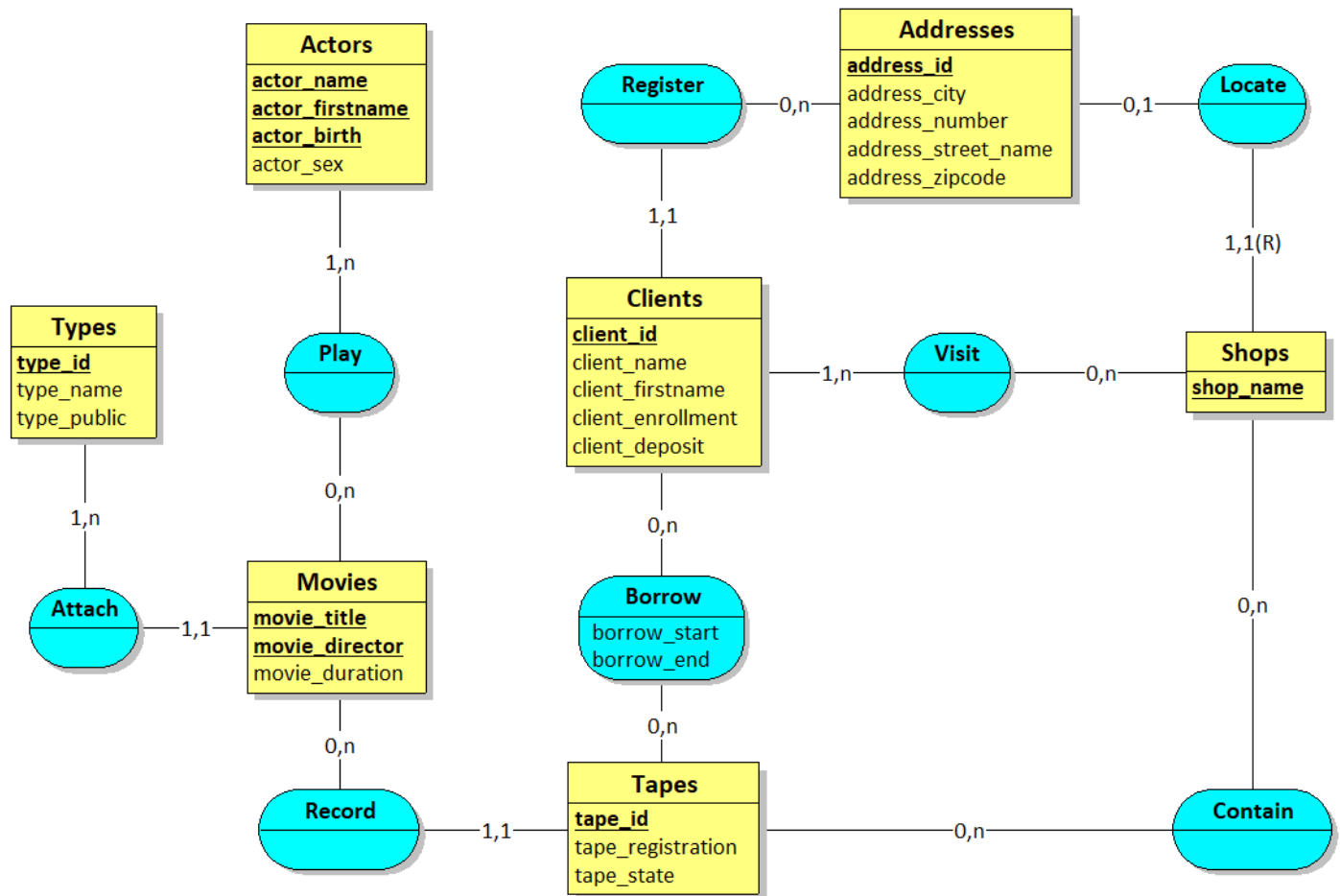
actor name, actor firstname, actor birth, movie title, movie director → actor_sex, movie_duration

tape id, client id → tape_registration, tape_state, client_name, client_firstname,
client_enrollment, client_deposit, borrow_start, borrow_end

tape id, shop name, address id → tape_registration, tape_state, address_zipcode, address_city,
address_number, address_street_name

client id, shop name, address id → client_name, client_firstname, client_enrollment,
client_deposit, address_zipcode, address_city, address_number, address_street_name

Modèle conceptuel de données





Modèle logique de données

Types = (**type id**, type_name, type_public);

Movies = (**movie title**, **movie director**, movie_duration, #type_id);

Actors = (**actor name**, **actor firstname**, **actor birth**, actor_sex);

Tapes = (**tape id**, tape_registration, tape_state, #(movie_title, movie_director));

Addresses = (**address id**, address_city, address_number, address_street_name, address_zipcode);

Clients = (**client id**, client_name, client_firstname, client_enrollment, client_deposit, #address_id) ;

Shops = (**shop name**, **#address id**);

Visit = (**#client id**, **#(shop name, #address id)**);

Contain = (**#tape id**, **#(shop name, #address id)**);

Play = (**#(actor name, actor firstname, actor birth)**, **#(movie title, movie director)**);

Borrow = (**#client id**, **#tape id**, borrow_start, borrow_end);



Crédits

Aurelien BOUDIER : Dieu

Jonathan JEANNIARD : Demi-dieu

Timothy BROWN : Jeune Padawan

Reda HAMZA : Vieux Padawan mais toujours pas Yoda