

Modélisation pour le projet de BD6

Emile ROLLEY

Remy PHOL-ASA

2020/2021

Abstract

Ce document contient la modélisation d'une base de donnée d'un site de e-commerce pour photographe.

Contents

Choix des produits	1
Diagramme E/R	2
Traduction en un schéma relationnel	3
Restructuration des spécialisations	3
Suppressions des relations	5
Schéma relationnel	6

Choix des produits

Nous avons décidé que notre site de e-commerce permettrait à des photographes de vendre leurs photographies.

Les client·e·s peuvent acheter les copies de photographie originales. Elles sont disponibles en deux formats (inclusifs) : numérique ou papier. Nous avons fait le choix de rendre les **copies numérique** comme étant **toujours disponibles**. Au **contraire des versions imprimées** qui doivent être disponibles pour pouvoir être commandées. De plus, **seules les copies imprimées sont retournables**.

Diagramme E/R

La première étape la modélisation consiste à la création d'un diagramme E/R :

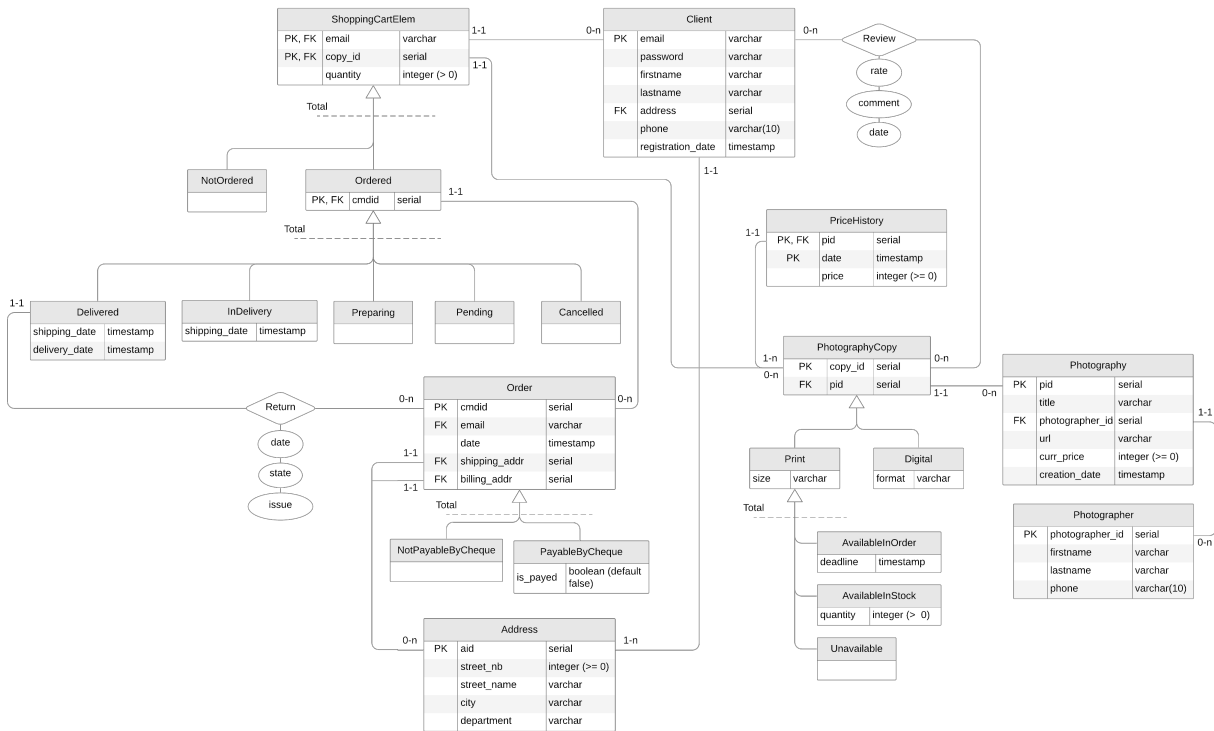


Figure 1: Diagramme E/R.

Les relations non nommées sont uniquement là pour montrer les cardinalités des couples (PK, FK), où PK signifie Primary Key et FK signifie Foreign Key.

Complété des contraintes externes suivantes :

- Un client (Client) peut émettre un commentaire (Review) sur une copie de photographie (PhotographyCopy) uniquement si il-elle l'a déjà reçu (Delivered).
- Seuls les produits délivrés (Delivered) peuvent être retournés (Return).
- La date d'expédition doit être inférieure à la date de réception : `Delivered.shipping_date < Delivered.received_date`.
- Si l'adresse de facturation (`Order.billing_addr`) n'est pas renseignée, l'adresse d'expédition (`Order.shipping_addr`) est utilisée.
- Un élément d'un panier (ShoppingCartElem) peut être annulé (Cancelled) seulement si il est en attente (Pending) ou en préparation (Preparing).
- Lorsqu'une copie est ajoutée dans un panier comme élément (ShoppingCartElem) : `AvailableInStock.quantity -= ShoppingCartElem.quantity`
- Lorsqu'un élément du panier (ShoppingCartElem) est annulé (Cancelled) : `AvailableInStock.quantity += ShoppingCartElem.quantity`
- Toute modification du prix d'une photographie (Photography) doit être reportée dans l'historique (PriceHistory).
- Seules les copies imprimées (Print) sont retournables (Returnable).
- Seuls les client-e-s (Client) connecté-e-s peuvent ajouter une copie (PhotographyCopy) dans leur panier (ShoppingCartElem).

Traduction en un schéma relationnel

Avant de pouvoir traduire notre diagramme E/R nous devons le restructurer en éliminant les spécialisations

Restructuration des spécialisations

La première étape consiste en la restructuration des spécialisations : PayableByCheque, NotPayableByCheque, Delivered, InDelivery, Preparing, Pending, Cancelled, Available, AvailableInStock et Unavailable.

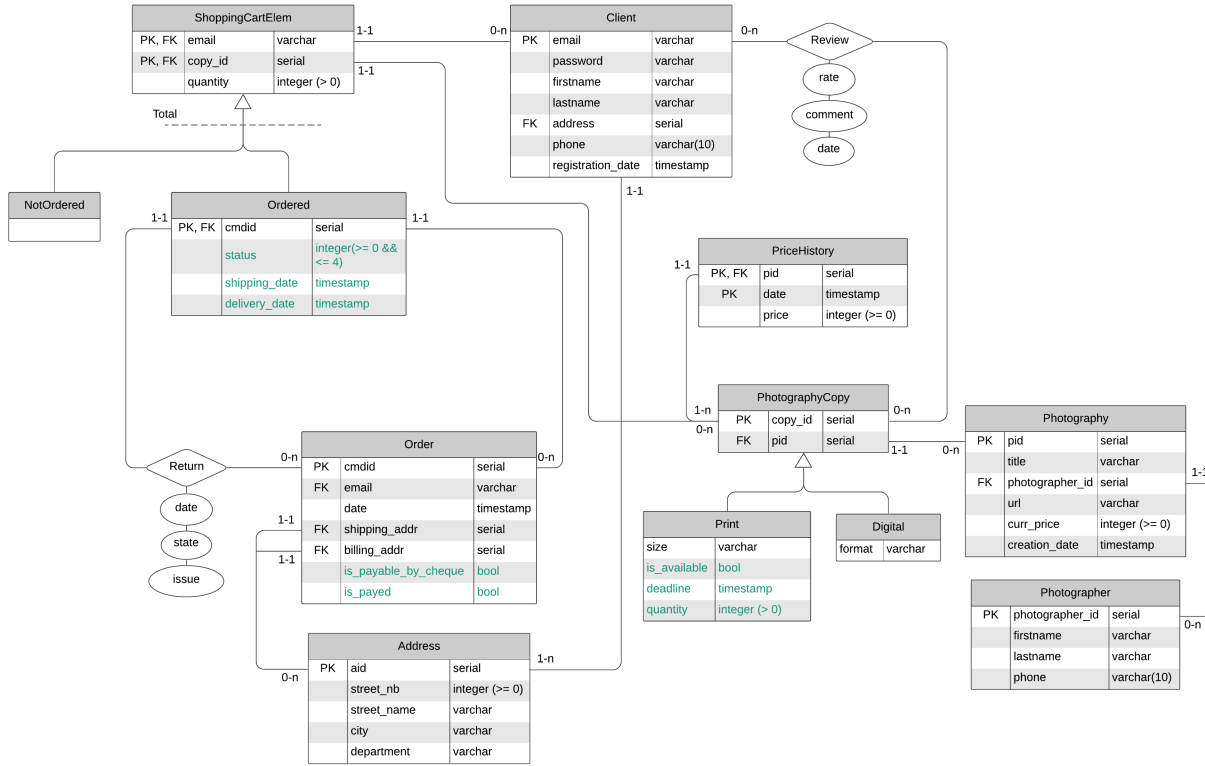


Figure 2: Diagramme E/R après la première restructuration (en vert les attributs ajoutés).

Les contraintes suivantes sont également ajoutées :

- Si `Print.is_available = true` alors si `Print.quantity > 0` la copie (`PhotographyCopy`) est considérée comme `AvailableInStock` sinon `Available`. De plus si `Print.is_available = false` alors la copie (`PhotographyCopy`) est considérée comme `Unavailable`.
- Significations des valeurs de `Ordered.status` :
 - 0 → Pending.
 - 1 → Preparing.
 - 2 → InDelivery.
 - 3 → Delivered.
 - 4 → Cancelled.

La deuxième étape permet la restructuration des spécialisations : **NotOrdered**, **Ordered**, **Print** et **Digital**.

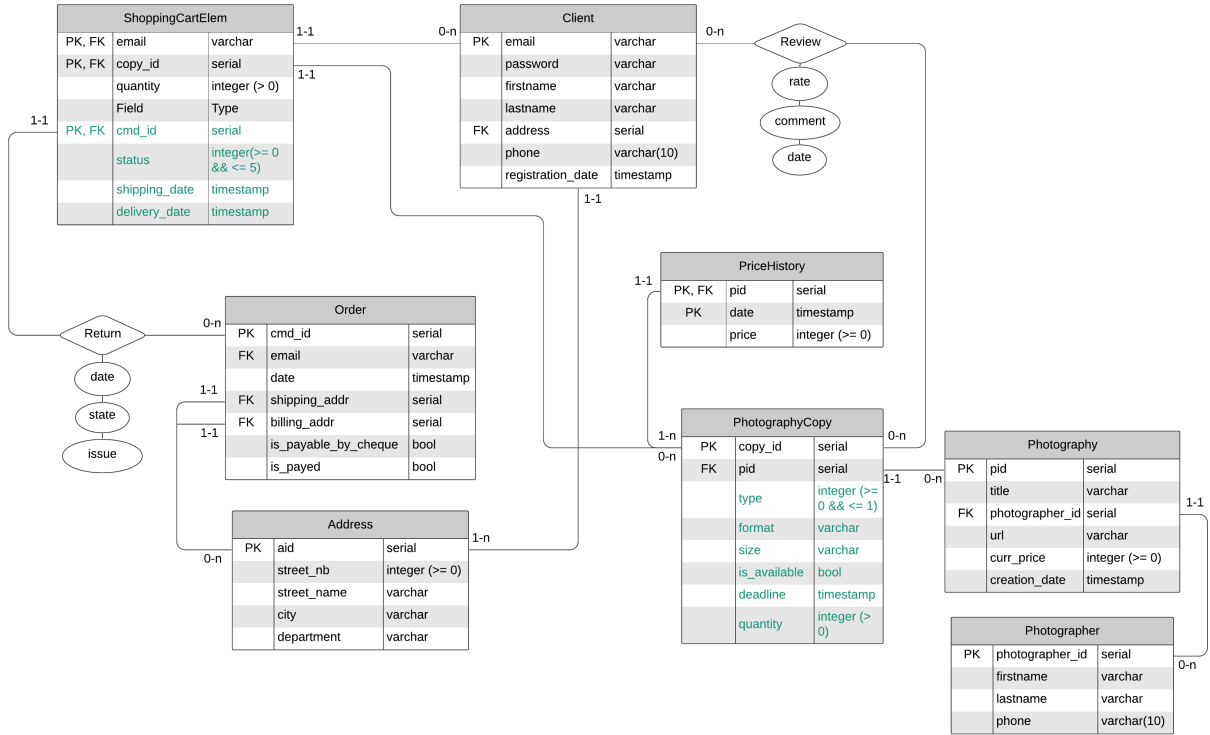


Figure 3: Diagramme E/R après la seconde restructuration (en vert les attributs ajoutés).

Les contraintes suivantes sont alors ajoutées :

- Significations des valeurs de **PhotographyCopy.type** :
 - 0 → **Print**.
 - 1 → **Digital**.
- Les valeurs de **ShoppingCartElem.status** possèdent les mêmes significations que pour **Ordered.status**, avec en plus :
 - [0, 4] → **Ordered**.
 - 5 → **NotOrdered**.

Suppressions des relations

Après la restructuration des spécialisations, nous pouvons supprimer les relations : **Return** et **Review**.

ShoppingCartElem		
PK	elem_id	serial
FK	email	varchar
FK	copy_id	serial
	quantity	integer (> 0)
FK	cmd_id	serial
	status	integer(>= 0 && <= 4)
	shipping_date	timestamp
	delivery_date	timestamp

Return		
PK, FK	elem_id	serial
PK, FK	cmd_id	serial
	date	timestamp
	issue	text

Client		
PK	email	varchar
	password	varchar
	firstname	varchar
	lastname	varchar
FK	address	serial
	phone	varchar(10)
	registration_date	timestamp

Order		
PK	cmd_id	serial
FK	email	varchar
	date	timestamp
FK	shipping_addr	serial
FK	billing_addr	serial
	is_payable_by_cheque	bool
	is_payed	bool

Address		
PK	aid	serial
	street_nb	integer (>= 0)
	street_name	varchar
	city	varchar
	department	varchar

Review		
PK, FK	email	serial
PK, FK	copy_id	serial
	rate	integer (>= 0 & <= 10)
	comment	text
	date	timestamp

PriceHistory		
PK, FK	pid	serial
PK	date	timestamp
	price	integer (>= 0)

Photography		
PK	pid	serial
	title	varchar
FK	photographer_id	serial
	url	varchar
	curr_price	integer (>= 0)
	creation_date	timestamp

PhotographyCopy		
PK	copy_id	serial
FK	pid	serial
	type	integer (>= 0 && <= 1)
	format	varchar
	size	varchar
	is_available	bool
	deadline	timestamp
	quantity	integer (> 0)

Photographer		
PK	photographer_id	serial
	firstname	varchar
	lastname	varchar
	phone	varchar(10)

Figure 4: Tables après la suppressions des relations.

Schéma relationnel

Finalement, nous avons le schéma relationnel suivant :

Photographer(photographer_id, firstname, lastname, phone)
Photography(pid, title, photographer_id, url, curr_price, creation_date)
 $Photography[photographer_id] \subseteq Photographer[photographer_id]$
PhotographyCopy(copy_id, pid, type, format, size, is_available, deadline, quantity)
 $PhotographyCopy[pid] \subseteq Photography[pid]$
PriceHistory(pid, date, price)
 $PriceHistory[pid] \subseteq Photography[pid]$
Address(aid, street_nb, street_name, city, departement)
Client(email, password, firstname, lastname, address, phone, registration_date)
 $Client[adresse] \subseteq Address[aid]$
Review(email, copy_id, rate, comment, date)
 $Review[email] \subseteq Client[email]$
 $Review[copy_id] \subseteq PhotographyCopy[copy_id]$
Order(cmd_id, email, date, shipping_addr, billing_addr, is_payable_by_cheque, is_payed)
 $Order[email] \subseteq Client[email]$
 $Order[shipping_addr] \subseteq Address[aid]$
 $Order[billing_addr] \subseteq Address[aid]$
ShoppingCartElem(elem_id, email, copy_id, quantity, cmd_id, status, shipping_date, delivery_date)
 $ShoppingCartElem[email] \subseteq Client[email]$
 $ShoppingCartElem[copy_id] \subseteq PhotographyCopy[copy_id]$
 $ShoppingCartElem[cmd_id] \subseteq Order[cmd_id]$
Return(elem_id, cmd_id, date, issue)
 $Return[elem_id] \subseteq ShoppingCartElem[elem_id]$
 $Return[cmd_id] \subseteq Order[cmd_id]$
