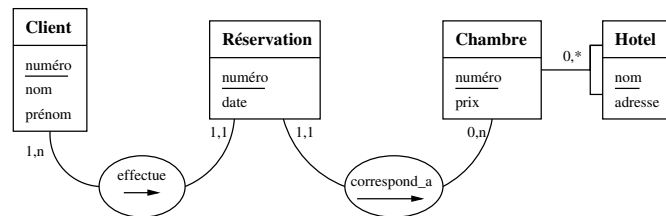


LIF4 - TD5 : Schémas entités associations

Correction

Exercice 1:

On souhaite gérer des réservations dans une compagnie d'hôtels. À cette fin, on considère le diagramme entité-association suivant (les attributs soulignés sont les identifiants des entités) :



1. À partir de ce diagramme, répondre aux questions suivantes:

(a) Est-il possible d'avoir des clients homonymes ?

Correction: oui, car le nom n'identifie pas les clients. Il peut donc y avoir des homonymes, ils auront des numéros différents.

(b) Un client peut-il réserver plusieurs chambres à une date donnée ?

Correction: oui, mais il faut faire plusieurs réservations

(c) Est-il possible de réserver une chambre sur plusieurs jours ?

Correction: oui, mais il faut faire plusieurs réservations

(d) Est-il possible de savoir si une chambre est libre à une date donnée ?

Correction: oui, il faut lister les réservations s'assurer qu'il n'y a aucune réservation à la date donnée.

(e) Est-il possible de réserver plusieurs fois une chambre à une date donnée ?

Correction: oui, rien ne garantit que la même chambre puisse être louée plusieurs fois.

2. Proposer un schéma de base correspondant au diagramme.

Correction:

- Client(numero,nom,prenom)
- Hotel(nom,adresse)
- Chambre(numero,prix,*nomhotel*)
- Reservation(numero,date,*numclient*, *nomhotel*,*numchambre*)

Penser à dire aux élèves qu'il est courant de mettre en italique les attributs d'une relation qui sont les identifiants d'une autre relation.

3. En utilisant les relations trouvées à la question 2, écrire des requêtes SQL pour:

(a) trouver les clients ayant réservé plusieurs chambres dans le même hôtel le 15 avril 2006;

Correction:

```

SELECT DISTINCT numclient
FROM Reservation
GROUP BY numclient, nomhotel
HAVING COUNT(*) > 1

```

- (b) donner les chambres disponibles le 15 avril 2006 à l'hôtel "Belaventure".

Correction:

```

SELECT numero
FROM Chambre
WHERE nomhotel='Belaventure'
AND numero NOT IN
  (SELECT numchambre
   FROM Reservation
   WHERE nomhotel='Belaventure'
   AND date = '15/04/2006')

```

Exercice 2:

Partie 1

Le directeur d'une chaîne de magasins de vente de CD (la "Fnak") vous charge de créer le système d'informations de son service de vente par Internet ("Fnak.com"). Afin de mieux cerner les besoins de cette entreprise, vous avez interrogé le directeur et pu en dégager les points suivants :

La seule activité de Fnak.com est la vente de CD musicaux au public français sur Internet.

Les visiteurs du site de vente en ligne ne peuvent acheter qu'après avoir ouvert un compte client auquel ils se connectent en entrant leur e-mail et un mot de passe d'au moins huit caractères. Lors de leur inscription, il leur est demandé leur état civil, leur adresse (celle qui sera utilisée pour la facturation des CD achetés) et leur numéro de carte de fidélité des magasins Fnak (s'ils en possèdent une).

Pour chaque CD du catalogue, un visiteur peut visualiser la pochette de l'album et un commentaire de cet album. Il peut également savoir si c'est une édition spéciale de l'album (un CD Collector par exemple). Le numéro de code barre d'un CD est différent selon l'édition de l'album.

Le prix (exprimé en euros) d'un CD dépend de la date depuis laquelle il est sorti. En général, à sa sortie, son prix est modéré, puis quelques semaines plus tard, le prix est rehaussé. Enfin, les CD à succès sont vendus à prix cassé au bout de quelques mois.

Lors d'une visite sur le site, un client peut créer un nouveau "panier" dans lequel il met le ou les CD qu'il veut acheter ainsi que le nombre d'exemplaires qu'il souhaite acheter. Ce panier est modifiable tant que le paiement de celui-ci n'est pas validé et confirmé. Le paiement s'effectue par un intermédiaire spécialisé qui se charge de prélever le coût de leur commande aux clients, de confirmer à Fnak.com que la commande a bien été payée et d'éditer un numéro unique de paiement. Une facture est alors établie, chaque facture a un numéro (unique mais les numéros ne se suivent pas forcément), les coordonnées du client, une adresse de facturation, une adresse de livraison et la liste des CD achetés.

Il existe un outil permettant de rechercher un CD par son auteur, son titre ou le nom d'une piste de ce CD. A chaque CD consulté, une liste d'autres CD est proposée au client. Cette liste est du type "les internautes ayant acheté ce CD ont aussi acheté :".

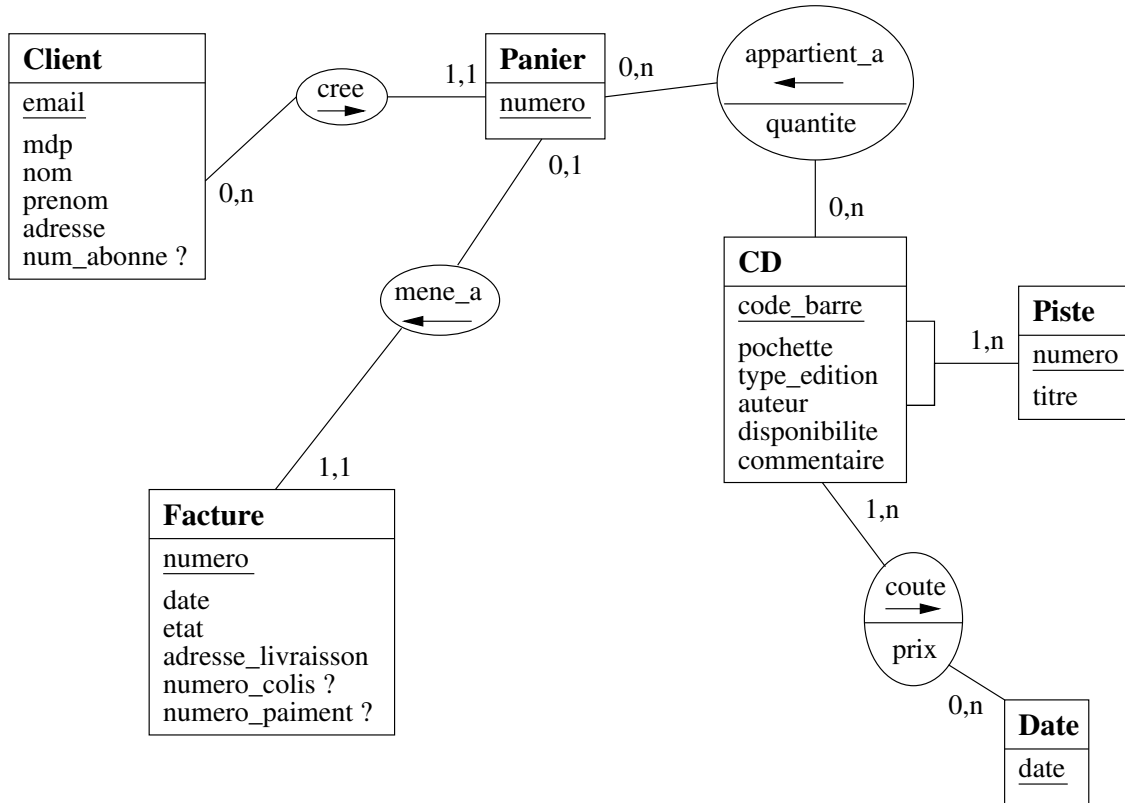
Des frais de livraison sont facturés aux clients : chaque commande coûte 3 euros plus 1 euro par CD commandé.

Une commande peut avoir plusieurs états après sa validation : "en attente de réapprovisionnement" lorsqu'un ou plusieurs CD n'est pas en stock, "en cours de traitement" lorsque la commande est en cours de traitement, "en cours de livraison" lorsqu'elle est chez le transporteur

(le client peut alors consulter le numéro du colis postal correspondant) puis finalement "livrée" lorsque le transporteur a confirmé que le colis a bien été livré.

Proposer un diagramme entité-association puis un schéma de base correspondant à cette description en français.

Correction:



Il existe sans doute plusieurs modèles possibles! En voilà un... J'ai ajouté quelques commentaires

- CLIENT(email, mdp, nom, prenom, adresse, num_abonne)
 - Les clients s’identifient grâce à leur email. L’email semble donc être un bon identifiant pour cette relation
 - L’adresse sera celle de facturation, une adresse de livraison différente pourra être utilisée pour chaque commande
 - num_abonne pourra être NULL
 - mdp doit faire au moins 8 caractères
- CD(code_barre, nom, auteur, pochette, commentaire, type_edition, disponibilite)
 - L’attribut pochette est une image (ou le nom d’un fichier image)
 - commentaire est un attribut de type texte long, il est raisonnable de penser que le commentaire dépend de l’édition de ce CD et pas seulement de l’album
 - type_edition est un attribut pouvant prendre les valeurs "Normal", "Collector", "Remasterisé"...

- disponibilite renseigne sur la quantité en stock
- **PRIX**(code_barre, DateDebutPrix, PrixEnEuro)
 - code_barre est une clé étrangère de la table CD
 - Pour connaître le prix d'un CD, il faut trouver le prix correspondant à la date d'achat, *i.e.*, le prix le plus récent par rapport à la date d'achat.
- **PANIER**(num_panier)
- **FACTURE**(num_facture, date_facture, adresse_livraison, num_paiement, etat, num_colis, num_panier, email)
 - date_facture correspond à la date à laquelle la commande a été réglée, elle peut donc être NULL tant que la commande n'est pas validée
 - L'adresse_livraison est saisie lors du paiement, elle peut donc être NULL tant que la commande n'est pas validée
 - email est l'identifiant d'un CLIENT
 - etat peut avoir pour statut "En cours de paiement" et toutes les valeurs de statut de l'énoncé
 - num_paiement (resp. num_colis) est NULL jusqu'à la validation du paiement (resp. l'envoi de la commande)
 - num_panier permet de savoir à quel panier correspond la facture
- **COMPOSITIONPANIER**(num_panier,num_code_barre, quantité)
 - num_panier est un identifiant d'un PANIER
 - num_code_barre est un identifiant de CD
 - quantité est le nombre d'exemplaires de num_code_barre commandés lors de du panier num_panier
- **COMPOSITIONCD**(num_code_barre, num_piste, titre)
 - num_code_barre est un identifiant de CD
 - Cette relation est indispensable pour la recherche des CD par nom de morceaux
- Les CD de type "les internautes ayant acheté ce CD ont aussi acheté :" ne nécessitent pas de nouvelle relation, tous les renseignements nécessaires à ce traitement sont dans la relation COMPOSITIONPANIER et FACTURE
- Les coûts de livraison sont des informations fixes qui ne varient en aucun cas. Il n'est donc pas nécessaire de les mémoriser dans notre système.
- Le montant des factures est calculable, il n'est donc pas nécessaire de le mémoriser. En pratique, une vue matérialisée serait utilisée pour accélérer la recherche du montant d'une facture et donc pouvoir calculer rapidement le chiffre d'affaire (par exemple). On est pas obligé d'en parler aux élèves cette année.
- En pratique, la gestion des stocks serait sans doute plus compliquée.
- Les tarifs postaux sont susceptibles de changer, il serait donc sans doute plus prudent d'en faire des informations dans notre base de données.

Partie 2

Quelques mois après l'ouverture du site *Fnack.com*, le succès est tel que le directeur souhaite proposer son service dans d'autres pays. Proposer une modification de votre schéma permettant de prendre en compte ce nouveau paramètre.

Tous les pays ne sont pas francophones, il faut donc traduire les informations et proposer à un client une version du site traduite dans sa langue maternelle.

Les monnaies ne sont pas les mêmes, le prix est donc calculé en fonction du taux de change (qui change tous les jours) et arrondi à une puissance de 10 dépendante de la monnaie utilisée (on arrondit par exemple à 10 yens japonais près mais à 0,01 dollars canadien près).

Les frais de livraison dépendent du pays. Le prix dépend du nombre de colis ou du nombre de CD contenus dans le colis ou d'une combinaison de ces deux critères.

Correction:

- Il faut ajouter une relation DEVISE(Nom, arrondi)
 - arrondi est un entier donnant la puissance de 10 à utiliser pour arrondir les prix
- Il faut ajouter une nouvelle relation. PAYS(nom, devise, prix_colis, prix_CD)
 - devise est un identifiant d'une DEVISE
- Il faut ajouter un attribut pays (resp. langue) à la table CLIENT. Cet attribut est un identifiant d'un PAYS
- Il faut ajouter une table TAUXDECHANGE qui donne le coefficient de change au jour le jour TAUXDECHANGE(Devise, Date, Coefficient)
- Le commentaire d'un CD doit être traduit. Il faut donc retirer l'attribut commentaire de la table CD et ajouter une relation COMMENTAIRE(num_code_barre, langue, commentaire)
 - num_code_barre est un identifiant de CD
- Il faut traduire les statuts et les type d'éditions ce qui nous oblige à créer deux autres relations. STATUTTRADUITS(StatutEnFrancais, langue, StatutTraduit) et TYPEEDITIONSTRADUITS(TypeEnFrancais, langue, TypeTraduit)

Le diagramme mis à jour donne:

