

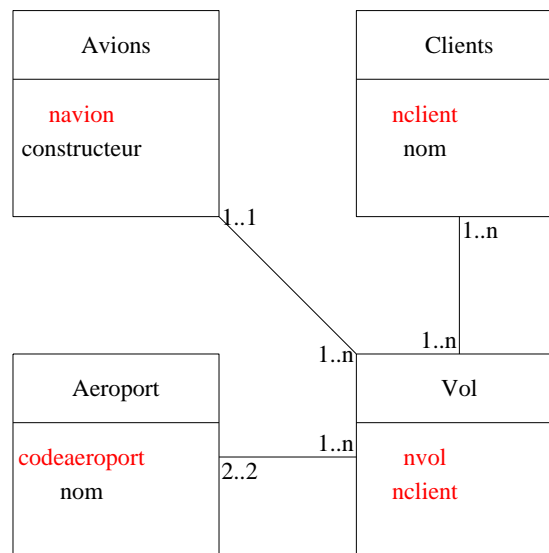
DS d'Informatique n° 2 : corrigé

PTSI Lycée Eiffel

9 mars 2018

Exercice 1

1. On va tout simplement créer une nouvelle entité appelée Vol. Pour ce qui est du choix de la clé primaire, on ne peut évidemment pas se contenter de repérer le vol par l'identifiant de l'avion effectuant le vol (un même avion va voler plusieurs fois), ni à l'aide de l'aéroport de départ (ou d'arrivée, ou même les deux) qui lui aussi ne sera pas unique. On ne peut même pas prendre une clé composée d'un couple identifiant d'avion et aéroport de départ, puisqu'un même avion peut très bien effectuer plusieurs fois le même trajet. À la limite, avoir le numéro de l'avion, l'aéroport de départ, celui d'arrivée et la date pourrait suffire, mais c'est nettement plus simple de créer tout simple un identifiant spécifique nvol. Pour les attributs on aura donc nvol (clé primaire), navion (le code d'identification de l'avion), aerdep (un code d'identification d'aéroport, pour le départ), aerarr (idem, pour l'arrivée), date (date du vol, par exemple au format jjmmaaaa).
2. Seuls les deux premiers attributs de chaque entité ont été indiqués :



3. On aura donc les relations suivantes (clé primaire en gras, clés secondaires en italique) :
 - Avions (**navion** integer, constructeur varchar(10), modele varchar(10), nbsieges integer)
 - Clients (**nclient** integer, nom varchar(20), prenom varchar(20), datenaissance date, nbpoints integer)
 - Aeroports (**codeaeroport** varchar(3), nom varchar(20), ville varchar(20), pays varchar(20))

- Vol (**nvol** integer, *navion* integer, *aerdep* varchar(3), *aerarr* varchar(3), *datevol* date)
- Embarquement (**nvol** integer, **nclient** integer, *nplace* integer)

On a créé une nouvelle entité pour faire le lien entre les vols et les clients, puisqu'il s'agit d'une relation de cardinalité 1..n 1..n. Cette nouvelle relation admet pour clé primaire le couple des clés primaires des relations Clients et Vol, et j'ai ajouté un attribut facultatif pour le numéro de place du client sur le vol.

- CREATE TABLE Aeroports (*codeaeroport* VARCHAR(3) PRIMARY KEY, *nom* VARCHAR(20), *ville* VARCHAR(20), *pays* VARCHAR(20))
- (a) $\Pi_{nom}(\sigma_{pays=Italie}(Aeroports))$
(b) $\Pi_{modele}(\sigma_{aerdep=CDG,date=12022007}(Avions \bowtie_{Avions.navion=Vol.avion} Vol))$
- (a) SELECT *nom*,*prenom* FROM Clients WHERE *nbpoints* > 500
(b) SELECT *nvol* FROM Vol, Avions WHERE Vol.*navion*=Avions.*navion* AND *modele*='A330' AND *date* LIKE '%062010'
(c) SELECT DISTINCT *modele* FROM Vol, Avions WHERE Vol.*navion*=Avions.*navion* AND *aerdep*=(SELECT *codeaeroport* FROM Aeroports WHERE *Ville*='Paris') AND *aerarr*=(SELECT *codeaeroport* FROM Aeroports WHERE *Ville*='Londres')
La commande DISTINCT, qu'on n'avait pas vue en cours et qui était donc évidemment facultative, permet d'afficher une seule fois chaque modèle. Je n'ai affiché que les liaisons où Paris était la ville de départ du vol.
(d) SELECT AVG(*nbpoints*) FROM Clients WHERE *nom* LIKE 'L%'
(e) en fait, on ne peut pas faire cette recherche facilement avec ce qu'on a vu en cours ! Pour se contenter d'afficher les aéroports espagnols ayant accueilli des vols en provenance de Bordeaux en 2017 :
SELECT *nom* FROM Vol, Aeroports WHERE Vol.*aerarr*=Aeroports.*codeaeroport* AND *date* LIKE '%2017' AND *aerdep*=(SELECT *codeaeroport* FROM Aeroport WHERE *Ville*='Bordeaux')

Exercice 2

- Non, puisque la clé primaire de la relation Achats, constituée du couple (Codemagasin,Codeclient), doit être unique, on ne peut donc pas avoir deux fois le même code de magasin et le même code client.
- ALTER TABLE Achats ADD *Dateachat* date
La clé primaire sera désormais un triplet (Codemagasin,Codeclient,*Dateachat*). Plus simplement, on pourrait d'ailleurs créer un code indépendant pour cette table.
- (a) SELECT *Nom* FROM Magasins WHERE *NbreEmployes* >4
(b) SELECT COUNT(*Nom*) FROM Clients WHERE *Nationalite*='Chinois'
(c) SELECT AVG(*Montant*) FROM Achats, Magasins WHERE Achats.*Codemagasin*=Magasins.*Codemagasin* AND *Nom*='Disney Store'
(d) SELECT *Nom* FROM Clients, Achats WHERE Achats.*Codeclient*=Clients.*Codeclient* AND *Montant* > 1000
On suppose que par montant total, il était sous-entendu le montant indiqué dans la table Achats, sinon on ne peut pas s'en sortir.