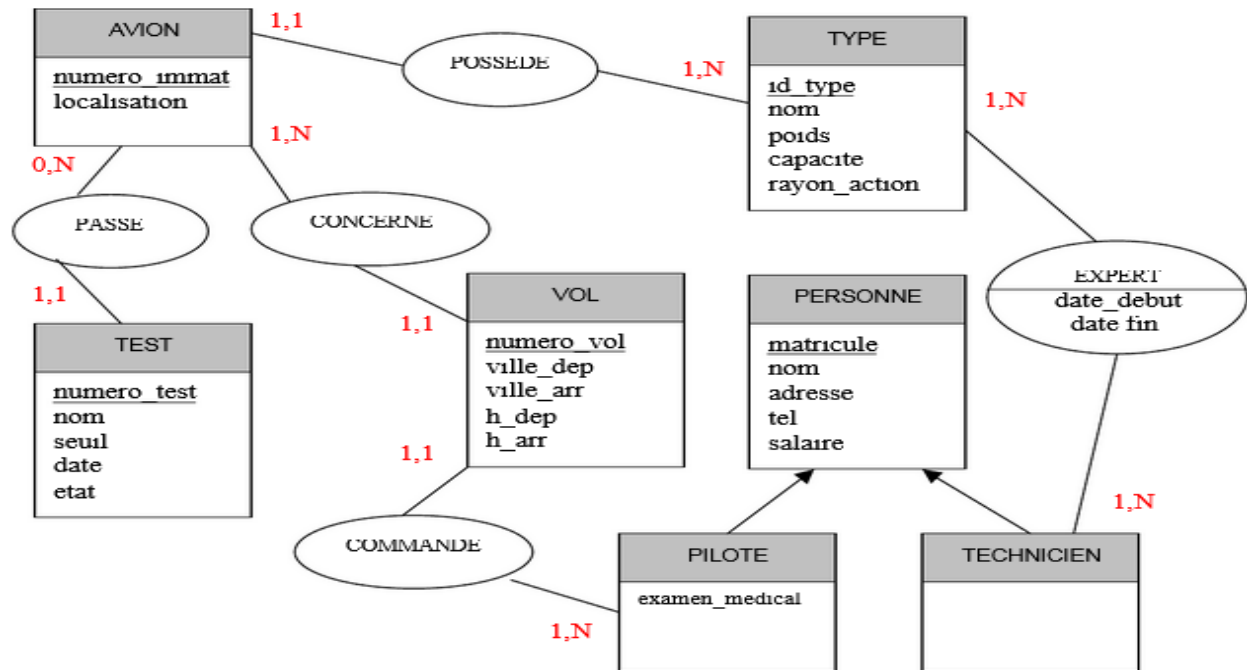


Corrigé des exercices de la série 1

Exercice 1 :



2. Traduire ce schéma conceptuel de données en schéma relationnel

AVION (numero_immat, localisation, # id_type)

TYPE (id_type, nom, poids, capacite, rayon_action)

EXPERTISE (#id_type, #matricule, date_debut, date_fin)

PERSONNE (matricule, nom, adresse, tel , salaire)

TECHNICIEN (matricule)

PILOTE (matricule, examen_medical)

OU

TECHNICIEN (matricule, nom, adresse, tel, salaire)

PILOTE (matricule, nom, adresse, tel, salaire, examen_medical)

OU

PERSONNE (matricule, nom, adresse, tel, salaire, examen_medical, type_personne) où
type_personne = TECHNICIEN ou PILOTE

VOL (numero_vol, # numero_immat, # matricule, ville_dep, ville_arr, h_dep, h_arr)

TEST (numero_test, nom, seuil, date, etat, # numero_immat)

3.

3.1. Donner la requête SQL de création de la table AVION

```
create table AVION (  
numero_immat varchar(20) not null,  
localisation varchar(50) not null,  
id_type integer not null,  
primary key (numero_immat),  
foreign key(id_type) references type(id_type)  
);
```

3.2. Donner la requête SQL de remplissage d'une ligne pour cette même table

```
insert into AVION values ( '1234XPZ', 'Bordeaux', 2);
```

Les numéros des pilotes en service et les villes de départ de leurs vols

- select matricule, ville_dep from vol ;

ou

- select vol.matricule, ville_dep
from vol, pilote
where vol.matricule = pilote.matricule ;

3.3. Le nom des pilotes domiciliés à Paris assurant un vol au départ de Nice avec un

Airbus A380

```
Select p.nom  
from pilote p, vol v, avion a, type t  
where p.matricule = v.matricule  
and v.numero_immat = a.numero_immat  
and a.id_type = t.id_type  
and adresse = 'Paris'
```

and ville_dep = 'Nice'

and type.nom = 'Airbus 380';

3.4. Les vols effectués par les pilotes de numéro 100 et 204

- select * from vol where matricule in (100, 204) ;

ou

- select * from vol where matricule = 100

union

select * from vol where matricule = 204;

4. par type d'avion, le nombre de vols au départ de Paris

- select t.nom, count(*)

from vol v, avion a, type t

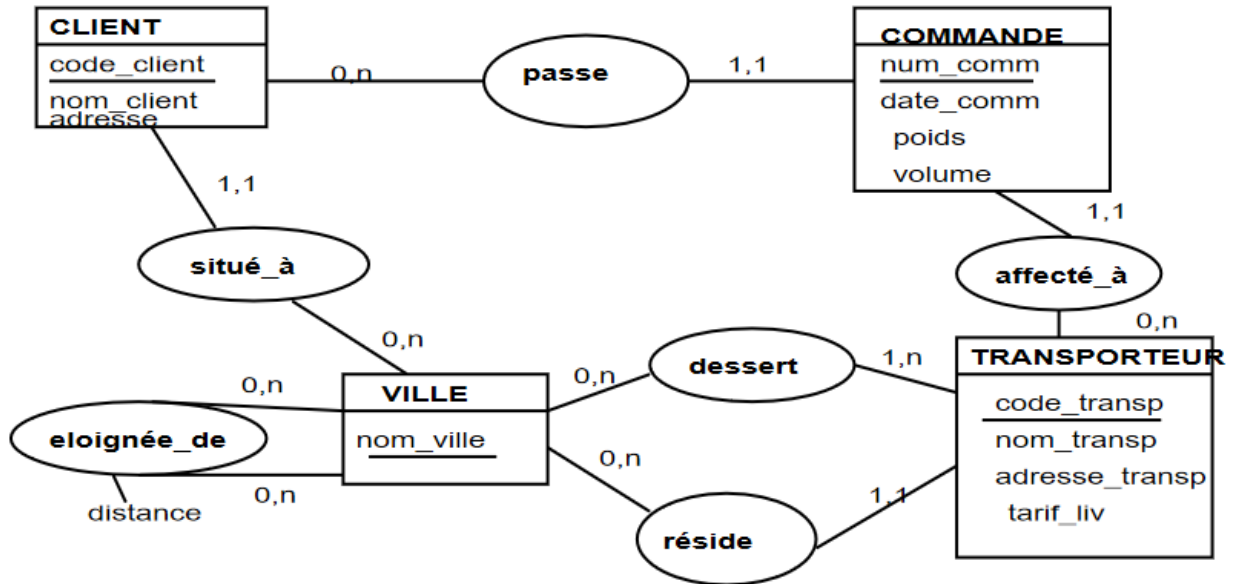
where a.id_type = t.id_type

and a.numero_immat = v.numero_immat

and ville_dep = 'Paris'

group by t.id_type ;

Exercice 2 :



Le modèle relationnel :

Les contraintes de domaines sont présentes dans le schéma relationnel de chaque relation.

Les contraintes d'intégrité de clés primaires sont lisibles dans les schémas (clé primaires soulignées).

Les types d'associations :

DESSERT(nom_ville:chaîne(30),code_transp:chaîne(10) ou nombre(10))

ELOIGNEE_DE(nom_ville1:chaîne(30), nom_ville2:chaîne(30), distance)

Les types d'entités :

VILLE(nom_ville:chaîne(30))

CLIENT(code_client:nombre(10), nom_client:chaîne(20), adresse:chaîne(30), nom_ville:chaîne(30))

COMMANDE(num_com:nombre(10), date_com:date, poids:nombre(4),volume:nombre(5), code_transp:nombre(10), code_client:nombre(10))

TRANSPORTEUR(code_transp:nombre(10), nom_transp:chaîne(20), adresse_transp:chaîne(30), tarif_liv:nombre(6), nom_ville:chaîne(chaîne(30)))

Contraintes d'intégrité référentielles (inclusion) :

code_transp de DESSERT est référencé par code_transp de TRANSPORTEUR

nom_ville de DESSERT est référencé par nom_ville de VILLE

nom_ville1 et nom_ville2 de ELOIGNEE_DE sont référencés par nom_ville de VILLE

nom_ville de CLIENT est référencé par nom_ville de VILLE

code_transp de COMMANDE est référencé par code_transp de TRANSPORTEUR

code_client de COMMANDE est référencé par code_client de CLIENT

nom_ville de TRANSPORTEUR est référencé par nom_ville de VILLE

Remarque : en pratique on peut ne pas implémenter la relation VILLE, ce qui supprime les contraintes d'inclusions associées.

Exercice 3 :

1. Une dépendance fonctionnelle DF établit d'abord une relation entre donnée, en plus d'être fonctionnelle.

- Matricule \rightarrow Nom, Age, signifie qu'il y a d'abord la relation "le matricule le nom et l'âge d'un certain élève" entre Matricule, Nom et Age. Ensuite, le nom et l'âge sont uniques pour un élève identifié par un matricule.
- Matricule \rightarrow Club, signifie un élève est "*inscrit*" ou "*participe*" à un club donné. En plus, ce club est unique (un élève ne participe pas à plus d'un club).
- Club \rightarrow Salle, signifie qu'un club "*a un local qui est une salle*". Cette salle est *unique*, (aucun club ne dispose de plus d'un local).

2. Si maintenant on considère la relation ELEVE (Matricule, Nom, Age, Club, Salle) on peut dire que l'attribut Matricule est clé, car il détermine tous les autres attributs, y compris Salle (la DF Matricule \rightarrow Salle est transitive).

Cette relation est en 2FN, car aucun attribut non clé ne dépend d'une partie de la clé (la clé n'est pas composée d'ailleurs).

Cette relation n'est pas en 3FN, car les attributs non clés ne sont pas mutuellement indépendants, à cause de la dépendance fonctionnelle Club \rightarrow Salle. (Une autre façon de dire, la DF par rapport à la clé Matricule \rightarrow Salle, n'est pas directe mais transitive par le fait que, par hypothèses, Matricule \rightarrow Club et que Club \rightarrow Salle).

Donc on projette la relation ELEVE pour que cette DF (Club \rightarrow Salle) soit due à une clé (Club).

ELEVE (Matricule, Nom, Age, Club)

ACTIVITE (Club, Salle)

qui sont deux relations en 3FN (car dans ELEVE, il n'y a plus de DFs entre attributs non clé). On retrouve la relation initiale par jointure de ces deux dernières relations, car Club est clé dans la deuxième relation.

Liens utilisés :

<https://vdocuments.mx/11565195-examen-correction-l2-base-de-donnees.html>

<https://cours.do.am/Kraya/exomldmcd.pdf>

<https://www.emi.ac.ma/ntounsi/COURS/DB/Polys/tdNormalisation-Corrige.pdf>

[SQL with Microsoft Access 2016 lesson 1 - Create table](#)