

--	--	--

**Bases de données – 2I009**

Examen du 18 Mai 2017

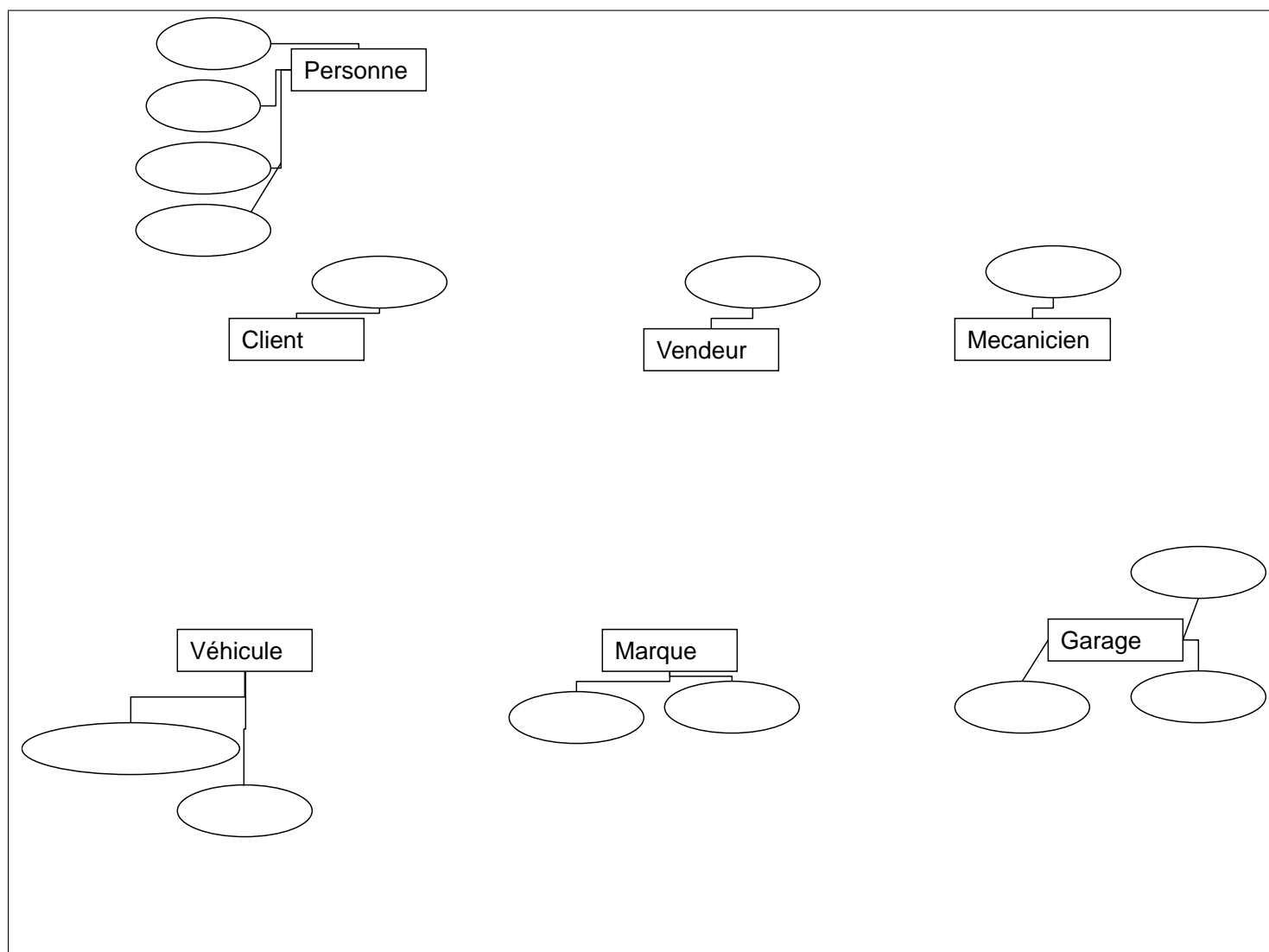
Durée : 2 heures – CORRIGÉ

Documents autorisés

*Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 20 points (14 questions) n'a qu'une valeur indicative.*

**1 Schéma Entité-Association (5 pts)**

On considère une base de données qui permet de gérer des garages avec leurs clients et qui est modélisée à l'aide d'un schéma entité-association dont **toutes** les entités apparaissent sur la figure suivante :



Chaque *Personne* est identifiée par un *idpers*. On connaît son *nom*, son *prénom* et son *numéro* de téléphone. Il y a trois types de personne dans la base : Clients, Vendeurs et Mécaniciens. Pour chaque client, on connaît son *adresse*. On connaît aussi le *niveau* professionnel de chaque mécanicien et le *taux* de commission (pourcentage sur le prix de vente d'un véhicule) de chaque vendeur.

Un *Véhicule* est identifié par son *immatriculation* et est d'un certain *modèle* et d'une certaine *marque*.

Une *Marque* d'un véhicule est identifiée par son *nom* ("Peugeot", "BMW", ...). On connaît la *ville* où se situe le siège social de la marque ("Sochaux", "Munich",...).

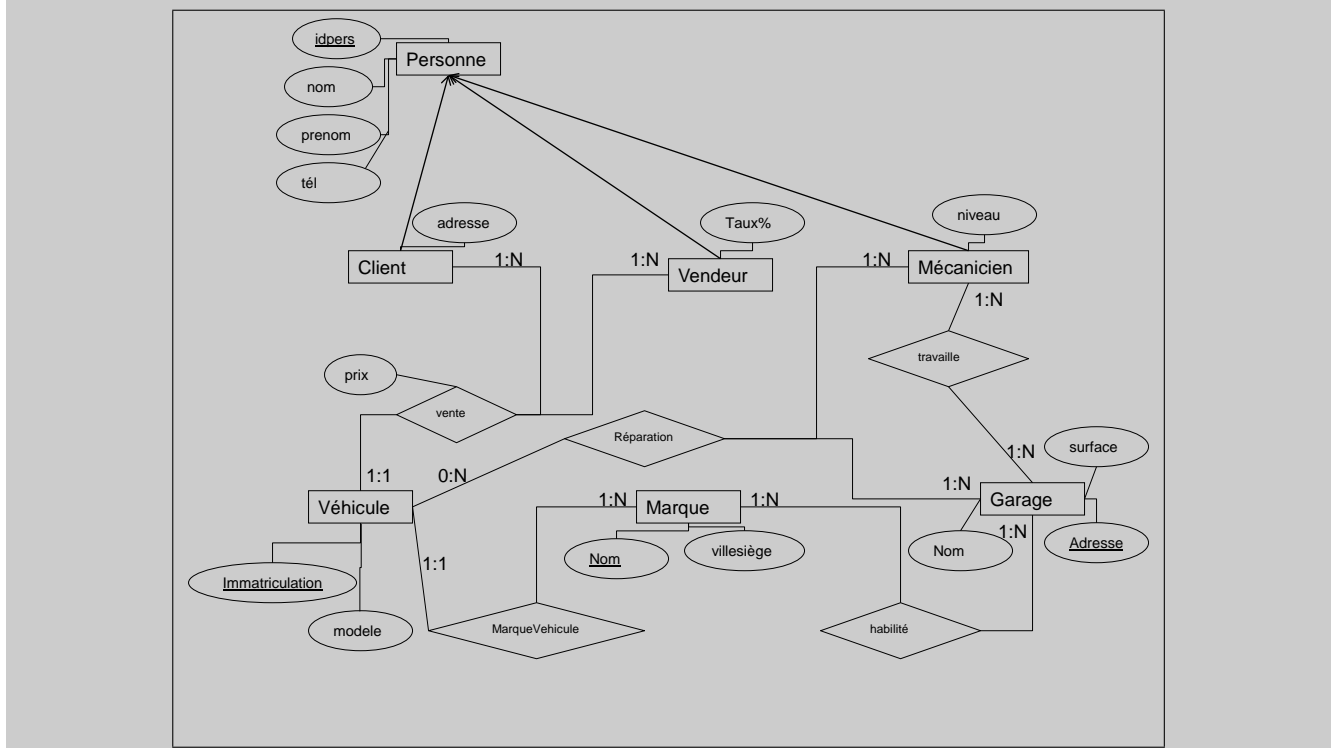
Un *Garage* a un *nom*, une *surface* et une *adresse*. Plusieurs garages peuvent avoir le même nom, mais il n'y a évidemment qu'un seul garage à une adresse donnée.

Un mécanicien travaille dans un ou plusieurs garages. Une réparation de véhicule est faite par un mécanicien dans un garage. Un véhicule est vendu (neuf) par un vendeur à un client à un certain prix. Un garage est habilité à faire des réparations sur les véhicules d'une ou plusieurs marques.

### Question 1 (3 points)

Complétez le schéma Entité-Association précédent. Ne pas oublier de préciser les attributs et les identifiants de chaque entité. Ajoutez également les associations correspondantes entre les entités en précisant leur **cardinalités** et leur éventuels **attributs**. *Rappel* : les liens de généralisation/spécialisation (héritage) sont représentés par des flèches, les entités faibles par un double rectangle, leur association identifiante par un double losange.

**Attention :** Vous ne pouvez pas rajouter des entités supplémentaires !

**Solution:**

**Question 2** (1½ points)

Compléter le cadre suivant pour traduire le schéma Entité/Association en relationnel. Pour chaque table, soulignez

les clés primaires et ajoutez une '\*' aux attributs qui sont des clés étrangères.

**NB : ne pas** donner les commandes de création de table et les types des attributs !

**Solution:**

Personne(idpers, nom, prénom, tél)

Vendeur(idpers, taux)

Mécanicien(idpers, niveau)

Client(idpers, adresse)

Garage(adresse, nom, surface)

Marque(nom, villesiège)

Véhicule(immatriculation, modèle, nommarque\*, idacheteur\*, idvendeur\*, prix)

Réparation(idmécanicien\*, immatvéhicule\*, adressegarage\*)

Travaille(idmécanicien\*, adressegarage\*)

Habilite(nommarque\*, adressegarage\*)

**Question 3** (1/2 point)

Quelle modification doit-on apporter au **schéma entité-association** si un mécanicien ne travaille que dans un seul garage ? Que cela change-t-il dans la modélisation des réparations ?

**Solution:** La cardinalité de Travailleur devient 1 :1 au niveau de Mécanicien. Du coup, l'association Réparation passe de ternaire à binaire (il suffit de lier véhicule à mécanicien, puisque le garage se déduit du mécanicien)

**2 Requêtes (10 pts)**

On considère le schéma relationnel suivant qui décrit des voyages :

**Destination** (nomD, pays)

**Voyage** (idV, type, nomD\*)

**Offre** (opérateur, idV\*, dateDép, prix, nbPlaces)

**Client** (nocli, nom, prénom, adresse)

**Réservation** (nocli\*, opérateur\*, idV\*, nbRes)

Les attributs de la clé primaire de chaque table sont soulignés et les attributs des clés étrangères sont indiqués par '\*' et portent le même nom que les attributs des clés primaires référencées. La table **Destination** contient la liste de destinations possibles où *nomD* est le nom de la destination (e.g "Baléares", "Crète", "Canaries", etc.) située dans un pays. Chaque voyage dans la table **Voyage** est identifié par *idV*, a un *type* (e.g "séjour", "club", etc), pour chaque voyage on connaît sa destination *nomD*. La table **Offre** stocke la liste de voyages proposés par chaque tour *opérateur* (e.g "Marmara", "Fram", "Look Voyages", etc.). Le même voyage peut être proposé par plusieurs opérateurs, pour chaque offre on connaît sa date de départ *dateDép*, le **prix par personne** ainsi que le nombre total de places *nbPlaces*. La table **Client** contient la liste de clients ayant réservé des voyages, chaque client étant identifié par son numéro *nocli*. La table **Réservation** contient pour chaque voyage réservé par un client auprès d'un opérateur, le nombre total de places réservées *nbRes*.

**Exprimez les requêtes ci-dessous dans les langages indiqués.**

**Question 1** (2 points)

Pour chaque client ayant réservé plus de 7 places pour un voyage à destination de "Crète" proposé par le tour opérateur "Héliades", donnez le nom du client ainsi que le **prix total** du voyage réservé.

**Solution:** Calcul.

$$\{c.nom, r.nbRes * o.Prix | Client(c) \wedge Reservation(r) \wedge Offre(o) \wedge r.nocli = c.nocli \wedge r.opérateur = o.opérateur \wedge r.idV = o.idV \wedge r.nbRes > 7 \wedge r.opérateur = "Heliades" \wedge \exists v(Voyage(v) \wedge v.idV = o.idV \wedge v.nomD = "Crete")\}$$

SQL :

```
select c.nom, r.nbres * o.prix
From Client c, Réservation r, Offre o, Voyage v
Where c.nocli = r.noCli and o.opérateur = r.opérateur
and o.idV = r.idV and v.idV=o.idV
and o.opérateur=' 'Héliades' ' and v.nomD=' 'Crete' '
and r.nbRes > 7
```

**Destination** (nomD, pays)

**Voyage** (idV, type, nomD\*)

**Offre** (opérateur, idV\*, dateDép, prix, nbPlaces)

**Client** (nocli, nom, prénom, adresse)

**Réservation**(nocli\*, opérateur\*, idV\*, nbRes)

### Question 2 (1 1/2 points)

Pour chaque destination proposée par le tour opérateur “Marmara”, donnez son nom (*nomD*) ainsi que le nom des autres tours opérateurs qui proposent cette destination **au même prix** par personne.

**Solution:** Calcul.

$$\{v.nomD, o2.opérateur | Voyage(v) \wedge Offre(o2) \wedge v.idV = o2.idV \wedge o2.opérateur \neq "Marmara" \wedge \exists o1(Offre(o1) \wedge o1.opérateur = "Marmara" \wedge o1.idV = v.idV \wedge o1.prix = o2.prix)\}$$

### Question 3 (1 1/2 points)

Les noms des tours opérateurs qui proposent **uniquement** des voyages à destination de “Madère”.

**Solution:** Calcul.

$$\{o.opérateur | Offre(o) \wedge \exists v(Voyage(v) \wedge v.idV = o.idV \wedge v.nomD = "Madere") \wedge \neg \exists o2, v2(Offre(o2) \wedge Voyage(v2) \wedge o2.idV = v2.idV \wedge o2.opérateur = o.opérateur \wedge v2.nomD \neq "Madere")\}$$

### Question 4 (1 point)

Le nombre de destinations en "Espagne" qui sont proposées par **un seul opérateur**.

**Solution:**

```
Select COUNT(*)
FROM Offre o, Voyage v, Destination d
Where o.idV = v.idV and v.nomD = d.nomD
d.pays=' 'Espagne' '
and NOT EXISTS (Select * FROM Offre o2, Voyage v2
Where o2.idV=v2.idV and o2.opérateur != o.opérateur and v2.nomD=v.nomD)
```

**Destination** (nomD, pays)

**Voyage** (idV, type, nomD\*)

**Offre** (opérateur, idV\*, dateDép, prix, nbPlaces)

**Client** (nocli, nom, prénom, adresse)

**Réservation**(nocli\*, opérateur\*, idV\*, nbRes)

### Question 5 (1 point)

Les dates de départ des voyages de type “séjour” qui n’ont plus de places disponibles (pour lesquels toutes les nbPlaces ont été réservées).

**Solution:**

```
select o.dateDép
from Voyage v, Offre o
```

```

where v.idV = o.idV and
v.type='séjour'
where nbPlaces =
(Select SUM (nbRes)
Réserve r
Where r.opérateur = o.opérateur and r.idV= o.idV)

```

**Question 6** (1 1/2 points)

Le nom de l'opérateur (des opérateurs) qui propose(nt) le plus grand nombre total de places (tous voyages confondus) à destination des "Canaries".

**Solution:**

```

select o.opérateur
from Offre o, Voyage v
where v.nomD = "Canaries"
and o.idV = v.idV
GROUP BY o.opérateur
HAVING SUM(nbPlaces) >= ALL (Select SUM(nbPlaces)
FROM Offre o2, Voyage v2
Where o2.idV = v2.idV
and v2.nomD = "Canaries"
and o2.opérateur != o.opérateur
GROUP BY o2.opérateur)

ou
HAVING SUM(nbPlaces) = (Select MAX(SUM(nbPlaces))...)

and o2.opérateur != o.opérateur pas nécessaire si >= ALL

```

**Destination** (nomD, pays)

**Voyage** (idV, type, nomD\*)

**Offre** (opérateur, idV\*, dateDép, prix, nbPlaces)

**Client** (nocli, nom, prénom, adresse)

**Réserve**(nocli\*, opérateur\*, idV\*, nbRes)

**Question 7** (1 1/2 points)

Les noms des opérateurs qui proposent des voyages vers **toutes les destinations** de la "Grèce".

**Solution:**

```

select o.opérateur
from Offre o where not exists (select * from Destination d
where d.pays = 'Grèce'
and not exists (select *
from Offre o2, Voyage v2
where o2.idV= v2.idV and v2.nomD=d.nomD
and
o2.opérateur = o.opérateur)

autre version possible en utilisant une agrégation

select o1.opérateur
from Offre o1, Voyage v1, Destination d1
Where o1.idV= v1.idV and v1.nomD=d1.nomD

```

```
and d1.pays = 'Grèce'
group by ol.opérateur
having count(d1.nomD) =
(select count(nomD) from Destination where pays = 'Grèce');
```

### 3 Création de tables, contraintes et mises à jour (5 pts)

On considère le schéma relationnel suivant où la clé primaire de chaque relation est soulignée et les attributs qui forment une clé étrangère sont suivis de \*. La base décrit des informations sur des articles, des clients qui achètent des articles et qui peuvent devenir VIP avec le temps.

**Articles** (aid, rayon, prix)

**Clients** (cid, nom, datenaiss)

**Achats** (aid\*, cid\*, dateachat, quantite)

**ClientsVIP** (cid\*, anciennete)

#### Question 1 (1 point)

Compléter les instructions de création de la table **Achats** et répondre à la question qui suit.

##### Solution:

```
create table achat (
aid varchar2(20), cid varchar2(10), dateachat date, quantite number,
primary key (aid, cid, dateachat),
foreign key (aid) references Articles
foreign key (cid) references Clients
```

un client peut acheter un même article plusieurs fois à des dates différentes car la clé primaire porte sur les trois attributs : aid, cid et dateachat.

**barème** : 1/4 pour PK, 1/4 pour les deux foreign key, 1/2 pour la réponse à la question.

#### Question 2 (1 point)

Les clients VIP qui ont acheté pour moins de 1000 Euros dans les 30 derniers jours doivent redevenir de simples clients (ne plus figurer dans la table ClientsVIP). Donner l'instruction qui permet de réaliser cette opération.

##### Solution:

```
delete from ClientsVIP
where cid in (select a.cid from Achats a, Articles r
              where a.aid=r.aid and sysdate - a.date <30
              group by a.cid
              having sum(r.quantite*a.prix)<1000
              and cid in (select cid from ClientsVIP));
```

**Attention** : la condition *cid in (select cid from ClientsVIP)* garantit que la suppression s'opère sur des clients de la table ClientsVIP. Je propose de ne pas pénaliser un tel oubli.

**Articles** (aid, rayon, prix)

**Clients** (cid, nom, datenaiss)

**Achats** (aid\*, cid\*, dateachat, quantite)

**ClientsVIP** (cid\*, anciennete)

**Question 3** (1 point)

On voudrait baisser de 20 % le prix des articles ayant été vendus à moins de 200 clients différents. Proposer une instruction qui permet de mettre à jour la table Articles en conséquence.

**Solution:**

```
update Articles
set prix = prix * .8
where aid in (select aid
              from Achats
              group by aid
              having count(distinct cid) <200);
```

**Question 4** (2 points)

On considère la table **ProduitsPourJeunes** dans laquelle on voudrait renseigner les articles achetés par au moins un client de moins de 26 ans et uniquement par des clients de moins de 26 ans. On voudrait également renseigner la moyenne d'âge des acheteurs de ces produits.

**ProduitsPourJeunes** (aid\*, MoyenneAge)

Proposer une instruction qui permet d'effectuer cette insertion. Ne **pas** donner l'instruction de création la table **ProduitsPourJeunes**

**Solution:**

```
insert into ProduitsPourJeunes
(select a.aid, avg((sysdate-c.datenaiss)/365)
from Achats a, Client c
where a.cid = c.cid and (sysdate-c.datenaiss)/365 <= 26
group by a.aid
);
```