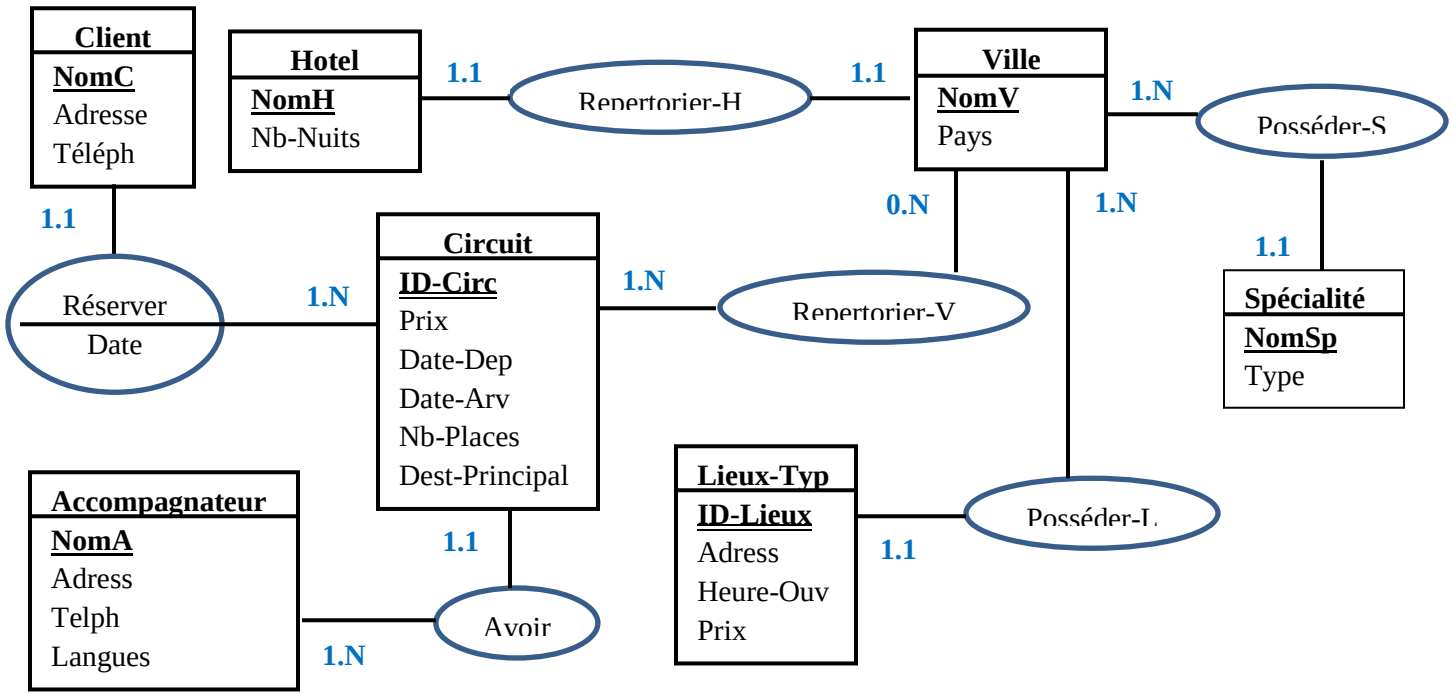
Université d'Alger 1. Département Mathématiques et Informatique
Module : Bases de Données
Année universitaire : 2016 /2017. L2 Informatique. Sections A et B.

Corrigé Type – Test 1 /15 pts

- **Exercice 1 : « Agence de voyages » : Proposition d'un Modèle E/A** **6.25 pts**



- **Exercice 2 : Du Modèle E/A au Modèle Relationnel** **8.75 pts**

  - **Les Règles de passage** (correspondantes au schéma E/A proposé)**: 1.5/8.75**

- **Règle 1-** Toute entité devient une relation ayant pour clé primaire son identifiant. Chaque propriété se transforme en attribut.

- **Règle 2-** Association plusieurs -à – plusieurs : transformer l'association en une table où la clé primaire est formée par la concaténation des identifiants des entités en relation. Toutes les propriétés de l'association deviennent des attributs qui ne peuvent pas faire partie de la clé.

- **Règle 3-** Association Un –à – Plusieurs : La clé primaire correspondant à l'entité côté n migre comme clé étrangère dans la relation correspondant à l'entité côté un.

  - **Le Modèle Relationnel : 2.25/8.75**

- **Recette** (**NomR**, Temps, Mode, **#NomRep**)
- **Ingrédient** (**NomI**, Type, Unité)
- **Contient** (**#NomR, #NomI**, quantité)
- **Repas** (**NomRep**, NBplats)

- **Les Requêtes SQL de création des tables : 5/8.75**

1- Table Repas: 1pt

```
CREATE TABLE Repas
( NomRep      CHAR(20)  NOT NULL,
  NBplat        NUMBER,
  CONSTRAINT clé_prim  PRIMARY KEY (NomRep));
```

2- Table Recette: 1.5pt

```
CREATE TABLE Recette
( NomR      CHAR(20)  NOT NULL,
  Temps      TIME,
   Mode       CHAR(20)/ NUMBER,
   NomRep  CHAR(20),
   CONSTRAINT clé_prim  PRIMARY KEY (NomR),
   CONSTRAINT clé_étrg  FOREIGN KEY (NomRep) REFERENCES Repas(NomRep));
```

3- Table Ingrédient: 1pt

```
CREATE TABLE Ingrédient
( NomI      CHAR(20)  NOT NULL,
  Type       CHAR(20),
  Unité      NUMBER,
   CONSTRAINT clé_prim  PRIMARY KEY (NomI));
```

4- Table Contient: 1.5pt

```
CREATE TABLE Contient
( NomR      CHAR(20)  NOT NULL,
  NomI       CHAR(20)  NOT NULL,
  quantité    NUMBER,
  CONSTRAINT clé_prim  PRIMARY KEY (NomR, NomI),
  CONSTRAINT clé_étrg1  FOREIGN KEY (NomR) REFERENCES Recette (NomR),
  CONSTRAINT clé_étrg2  FOREIGN KEY (NomI)  REFERENCES Ingredient (NomI));
```

Université d'Alger 1. Département Mathématiques et Informatique
Module : Bases de Données
Année universitaire : 2016 /2017. L2 Informatique. Sections A et B.

Corrigé Type – Test 2 /15 pts

Exercice 1 : On considère la base de données avion 7.5pts

PILOTE (**NUMPIL**, NOM_PIL, PRENOM_PIL, VILLE, SALAIRE)

AVION (**NUMAV**, NOM_AV, CAPACITE)

VOL (**NUMVOL**, NUMPIL, NUMAV, VILLE_DEP, VILLE_ARR, Heure_DEP, Heure_ARRV)

- **Exprimer les requêtes suivantes en Algèbre relationnelle**

0.5 1- Noms et prénoms des pilotes qui résident à Alger.

$\Pi_{NOM\_PIL, PRENOM\_PIL}(\sigma_{VILLE = 'Alger'}(PILOTE))$

1.5 2- Noms, Prénoms et villes des pilotes qui n'ont fait aucun vol.

$R1 = \Pi_{NUMPIL}(PILOTE) - \Pi_{NUMPIL}(VOL)$

$R2 = R1 \bowtie_{R1.NUMPIL = P.NUMPIL} (PILOTE)$

$R3 = \Pi_{NOM\_PIL, PRENOM\_PIL, VILLE}(R2)$

0.75 3- Les numéros d'avions de capacité plus que 500 places ou de ville de départ est Oran.

$\Pi_{NUMAV}(\sigma_{CAPACITE>500}(AVION)) \cup \Pi_{NUMAV}(\sigma_{VILLE\_DEP='Oran'}(VOL))$

1.5 4- Noms et prénoms des pilotes qui ont participé à tous les vols.

$R1 = \Pi_{NUMPIL, NUMVOL}(VOL) \div \Pi_{NUMVOL}(VOL)$

$R2 = R1 \bowtie_{R1.NUMPIL = P.NUMPIL} (PILOTE)$

$R3 = \Pi_{NOM\_PIL, PRENOM\_PIL}(R2)$

0.75 5- Numéros des pilotes de salaire moins de 2000 euro et qui n'ont pas participé au vol numéro 4

$\Pi_{NUMPIL}(\sigma_{SALAIRE<2000}(PILOTE)) \cap \Pi_{NUMPIL}(\sigma_{NUMVOL <> 4}(VOL))$

2.5 6- Noms des pilotes et noms des avions qu'ils n'ont pas utilisés dans un vol.

$R1 = \Pi_{NUMPIL}(PILOTE) \times \Pi_{NUMAV}(AVION)$

$R2 = R1 - \Pi_{NUMPIL, NUMAV}(VOL)$

$R3 = R2 \bowtie_{R1.NUMPIL = P.NUMPIL} (PILOTE)$

$R4 = R3 \bowtie_{R3.NUMAV = A.NUMAV} (AVION)$

$R5 = \Pi_{NOM\_PIL, NOM\_AV}(R4)$

Exercice 2 : On considère la base de données *course* 7.5pts

COUREUR (**numLicence**, Nom, Prénom, DateNaissance)
COURSE (**numCourse**, Ville, CodePostal)
RESULTAT (***numCourse, numLicence***, temps, rang)

- **Exprimer les requêtes suivantes en SQL**

1.5 1- les coureurs qui ne participent que dans les courses organisées à Alger

```
SELECT *  FROM COUREUR
WHERE numLicence NOT IN
(SELECT numLicence  FROM RESULTAT  R, COURSE C
WHERE R. numCourse = C. numCourse AND C.Ville <> "Alger");
```

1.25 2- les numéros de licence de coureurs effectuant un temps supérieur au coureur de licence 1

```
SELECT R1.numlLicence
FROM RESULTAT R1, RESULTAT R2
WHERE  R1.Temps  >  R2.Temps AND R2. numlLicence= 1;
```

1.75 3- les noms des coureurs nés avant 1980 et effectuant un temps minimal moins que 300sec triés par temps minimal croissant.

```
SELECT nom, MIN (temps)
FROM COUREUR, RESULTAT
WHERE coureur.numLicence = resultat.numLicence AND DateNaiss < 1980
GROUP BY nom
HAVING MIN (temps) < 30
ORDER BY MIN (temps);
```

2.5 4- Définir une vue « ***Course-Result*** » qui va contenir Numéro de licence, nom, prénom, numéro des courses, la ville et temps des coureurs où leur temps dans le rang 1 est inférieur à tous les temps du rang 2

```
CREATE VIEW Course-Result ( Num, NomC, PrenomC, CourNum,
CourVille, temp) AS
SELECT numLicence, Nom, Prénom, numCourse, Ville,  temps
FROM COUREUR C, COURSE S, RESULTAT R
WHERE  C. numLicence = R. numLicence AND
R. numCourse = S.numCourse AND rang=1 AND
R.temps < ALL ( SELECT Temps FROM  RESULTAT WHERE rang = 2)
```

0.5 5- Modifier dans la vue ***Course-Result*** la ville de la course numéro 1 à Alger.

```
UPDATE  Course-Result  SET ville = ''Alger'' WHERE CourNum=1;
```