



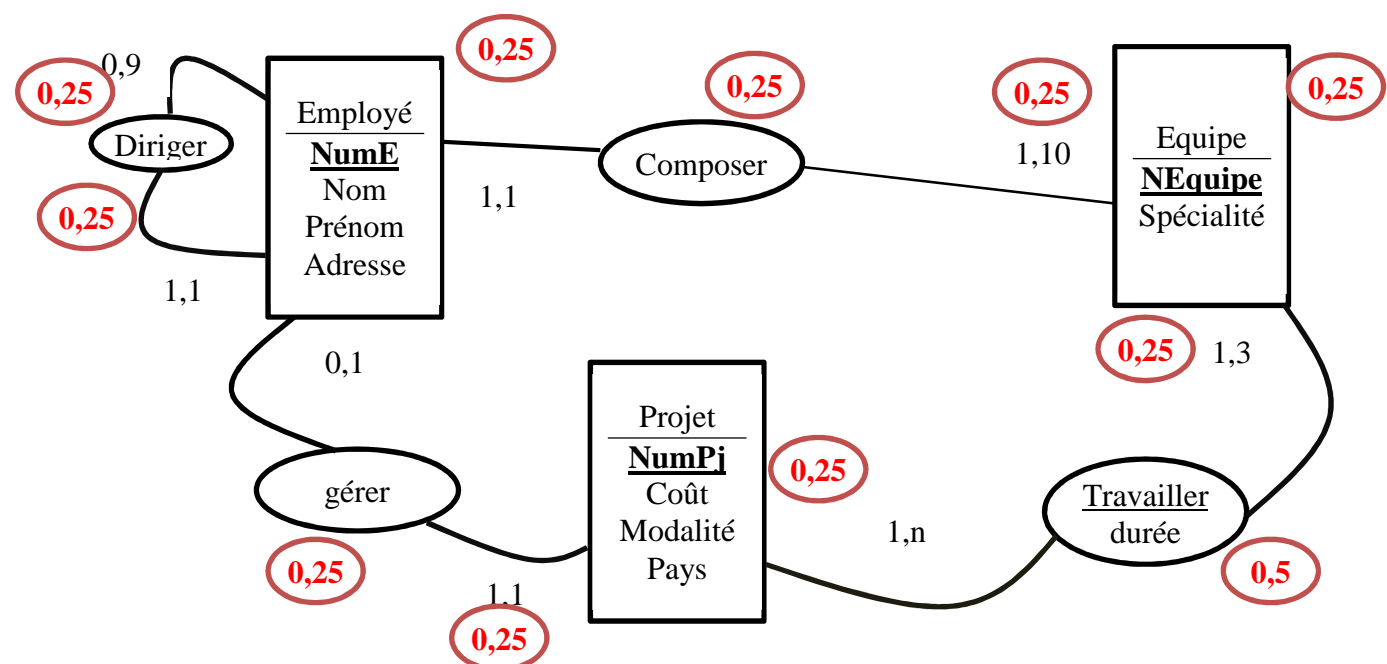
Corrigé type_ Examen - Base de données

Questions : (2 pts)

- 1-Définir une information ? Une combinaison de données dans un contexte bien définit. **0,25**
- 2-Quels sont les étapes essentielles vues en cours pour créer une base de données ? trois étapes essentielles:
- la modélisation conceptuelle (E/A), **0,25**
 - la modélisation logique(Relationnel), **0,25**
 - et la réalisation physique de la base (Requêtes SQL). **0,25**
- 3- Définir la modélisation ?
- Exprimer une situation comprenant des informations de la réalité de ce monde en un diagramme facile à comprendre, ce dernier ensuite peut facilement permettre le stockage physique des informations qu'il modélise. **0,25**
- 4- Définir la notion de vue dans une base de données ?
- Une vue est une table virtuelle calculée à partir des tables de base par une requête. **0,25**
- 5-Quels est l'intérêt de la normalisation ?
- Eviter la **redondance** de données tout en **gardant la cohérence** de la base de données. **0,5**

Exercice1 : (3pts)

Une société d'Architecture innoArchi veut améliorer sa gestion de suivi de projet en bâtiment. La société est composée de plusieurs équipes chaque équipe est composée de 10 employé au maximum. Une Equipe est caractérisé par une spécialité en bâtiment. Chaque équipe a un chef d'équipe qui fait partie des dix employés. La société a plusieurs projets de suivi localisés dans différents pays. Un projet est caractérisé par un nom, le coût et les modalités. A chaque projet innoArchi choisi un chef de projet parmi ses employés pour le diriger, un employé dirige au maximum un projet. Plusieurs équipes peuvent travailler sur un même projet et une équipe peut travailler au maximum sur trois projets en même temps. Donner le modèle entité/Association correspondant.





Corrigé type_ Examen - Base de données

Exercice 2

- 1-ASSOC_SPORTIVE(NumASSOC, NomASSOC)
- 2-DISCIPLINE (NumDiscip, Nomdiscipline)
- 3- Sportif(NumSportif, Nom, Prénom, #numDiscip_S)
- 4-Affiliation(#NumSportif A,# NumASSOC A)
- 5-Compétition (NumCompet, Date, lieu, nombre_Participant)
- 6-Participation(#NumSportif P, #NumCompet P)
- 7-Organisation (#NumASSOC_O, #NumCompet_O)

- 1- Donner le nom et le prénom des sportifs qui ont participé à toutes les compétitions organisés par l'association 'MedSport'. **1,25pt**

Les noms et prénom des sportifs pour qui, il n'y a aucune compétitions organisés par l'association MedSport au quels il n'a pas participé.

```
SELECT DISTINCT `nom`,`Prenom` FROM `sportif` AS S, participation AS P WHERE P.NumSportif_P=
S.numSportif AND P.NumCompet_P NOT IN
(SELECT C1.NumCompet FROM participation AS P1, competition AS C1, organisation AS O1
,assoc_sportive AS AS1 WHERE
C1.NumCompet=O1.NumCompet_O AND O1.numASSOC_O = AS1.NumASSOC AND
AS1.NomASSOC<>'MedSPORT' AND C1.NumCompet=P1.NumCompet_P
AND P1.NumSportif_P NOT IN
( SELECT P2.NumSportif_P FROM participation AS P2 WHERE
P2.NumCompet_P=C1.NumCompet AND
P2.NumSportif_P=S.numSportif
)
)
```

Ou bien

Select Nom, Prénom From Sportive

```
Where NumSportif IN (Select NumSportif_A From Participation
Where NumCompet_P IN (Select NumCompet_O
From organisation, assoc_sportive
Where NumASSOC = NumASSOC_O
and NomASSOC = 'MedSport')
Group by (numsportif)
Having count(NumCompet_P) = (Select count(*)
From organisation, assoc_sportive
Where NumASSOC = NumASSOC_O
and NomASSOC = 'MedSport'))
```

- 2- Donner le nom des Associations qui ont organisés un nombre de compétitions supérieur aux nombre organisé par 'MedSport'. **1,25pt**

```
SELECT * FROM `assoc_sportive` AS AS1 WHERE AS1.`NumASSOC` IN
```



Corrigé type_ Examen - Base de données

```
(SELECT `NumASSOC_O` FROM organisation AS O1
group by O1.numASSOC_O
HAVING Count(O1.numASSOC_O) > (SELECT COUNT(O2.numASSOC_O)
FROM organisation AS O2
WHERE O2.numASSOC_O = (SELECT AS2.NumASSOC FROM
assoc_sportive AS AS2 WHERE AS2.NomASSOC='medSPORT'
GROUP BY O2.numASSOC_O )))
```

Ou bien

```
Select NomASSOC from ASSOC_SPORTIVE
where NumASSOC IN ( Select NumASSOC_O From organisation
Group by (NumASSOC_O)
Having count (NumCompet_O) > (Select count (*) From organisation,
assoc_sportive
Where NumASSOC = NumASSOC_O
and NomASSOC = 'MedSport'))
```

- 3- Donner la date et le lieu des compétitions qui ont été organisées par l'association 'MedSport' qui ont un nombre de participants supérieur à toutes les compétitions organisées par l'association 'InfoSanté'. **1,25pt**
- ```
SELECT `Date1`, `Lieu`
FROM competition AS C1
WHERE C1.NumCompet IN (SELECT O1.NumCompet_O FROM organisation AS O1, assoc_sportive AS AS1
WHERE O1.numASSOC_O= AS1.NumASSOC AND AS1.NomASSOC='medSPORT')
AND C1.Nbr_Participant > ALL (SELECT C2.Nbr_Participant FROM competition AS C2
WHERE C2.NumCompet IN
(SELECT O2.NumCompet_O FROM organisation AS O2, assoc_sportive AS AS2 WHERE
O2.numASSOC_O= AS2.NumASSOC AND AS2.NomASSOC='InfoSanté'))
```

Ou bien

```
Select Date1, lieu
From Competition C1, organization O1, assoc_sportive ASS1
Where C1. NumCompet = O1. NumCompet_O and ASS1.NumASSOC = O1.NumASSOC_O
and ASS1.NomASSOC = 'MedSport' and C1.nombre_Participant > All (Select C2.nombre_Participant
From Competition C2, organization O2, assoc_sportive ASS2
Where C1. NumCompet = O1. NumCompet_O and ASS2.NumASSOC =
O2.NumASSOC_O
and ASS2.NomASSOC = 'InfoSanté')
```

- 4- Créer une vue qui permet d'avoir les nom et prénom et discipline des sportifs qui ont participé à la compétition du '18/07/2018' dont l'une des associations organisatrices est 'MedSport' **1,25pt**

Create view VUE1 AS



## Corrigé type\_ Examen - Base de données

```
(SELECT S.nom, S.Prenom, D.NumDiscip FROM sportif S, discipline D
WHERE S.numDiscip_S= D.NumDiscip and S.numSportif IN (SELECT P.NumSportif_P
FROM participation P, competition C, organization O, assoc_sportive ASS
WHERE P.NumCompet_P= O.NumCompet_O and O.NumCompet_O=C.NumCompet
and O.numASSOC_O= ASS.NumASSOC and ASS.NomASSOC='medsport' and
C.Date='2018-07-18'))
```

Ou bien

Create view VUE1 AS

```
(SELECT S.nom, S.Prenom, D.NumDiscip
FROM sportif S join discipline D
On S.numDiscip_S= D.NumDiscip
WHERE exist (SELECT *
FROM participation P, competition C, organization O, assoc_sportive ASS WHERE
P.NumCompet_P= O.NumCompet_O and O.NumCompet_O=C.NumCompet and
O.numASSOC_O= ASS.NumASSOC
and ASS.NomASSOC='MedSport' and C.Date='2018-07-18'
and P.NumSportif_P = S. NumSportif))
```

**Algèbre relationnelle :**

- a- Dates et lieux des compétitions organisées par les associations sportives qui organisent toutes les disciplines **(1.25)**

$\Pi \text{ date, lieu } [ [ \Pi \text{ NumASSOC, NumDiscip (Association} \bowtie \text{Affiliation} \bowtie \text{Sportif)} \div \Pi \text{ NumDiscip (Discipline)} ] \bowtie \text{Organisation} \bowtie \text{Compétition} ]$

- b- Les numéros de disciplines qui n'ont pas été pratiquée par aucun sportif ou qui ont été pratiquée par des sportifs qui ne sont affiliés à aucune association sportive **(1.25)**

$\Pi \text{ NumDiscip (Discipline)} - \Pi \text{ NumDiscip } [(\text{Sportif})] \text{ UNION}$

$\Pi \text{ NumDiscip (Discipline)} - \Pi \text{ NumDiscip } [ [ \Pi \text{ NumSportif (Sportif)} - \Pi \text{ NumSportif (Affiliation)} ] \bowtie \text{Sportif} ]$

### Résultats des requêtes :

- a-  $\sigma \text{ NumASSOC\_A} = 2 (\text{Affiliation}) \times [ (\Pi \text{ NumSportif, NumDiscip (Sportif)} \bowtie (\Pi \text{ NumSportif (Sportif)} - \Pi \text{ NumSportif (Affiliation)})) ]$  **(1.25)**



## Corrigé type\_ Examen - Base de données

| NumSportif_A | NumASSOC_A | NumSportif | numDiscip |
|--------------|------------|------------|-----------|
| 1            | 2          | 2          | 5         |
| 1            | 2          | 5          | 3         |
| 3            | 2          | 2          | 5         |
| 3            | 2          | 5          | 3         |
| 4            | 2          | 2          | 5         |
| 4            | 2          | 5          | 3         |

**b-** Sportif  $\bowtie$  [ (( $\Pi$  NumSportif, NumDiscip (Sportif)  $\div$   $\Pi$  NumDiscip, (Dicipline))  
 $\bowtie$  Affiliation)  $\Join$   $\Pi$  NumASSOC, NumSportif ( $\sigma$  NomASSOC= 'Assoc\_R'  
 (ASSOC\_SPORTIVE)  $\bowtie$  Affiliation) ] **(1.25)**

| NumSportif | Nom | Prénom | numDiscip_S | NumASSOC_A |
|------------|-----|--------|-------------|------------|
| Table Vide |     |        |             |            |

### Exercice 3: Formes Normales

#### 1- Les dépendances fonctionnelles

**a-** "On peut déduire le nom de la machine à partir de son type."

numTypeMachine  $\rightarrow$  nomMachine **(0.5)**

**b-** "Un technicien identifié par son numéro possède un seul montant de prime par type de machine".

numTypeMachine, numTechnicien  $\rightarrow$  montantPrime **(0.5)**

**c-** "Un numéro de technicien est associe à un seul nom de technicien".

numTechnicien  $\rightarrow$  nomTechnicien **(0.5)**



---

## Corrigé type\_ Examen - Base de données

---

**2-** la clé minimale : (numTypeMachine, numTechnicien). (0.5)

A partir du numéro de la machine et le numéro du technicien on peut déduire tous les autres attributs.

**3-** quelle forme normale est la relation “Prime” en justifiant.

La relation est en première forme normale. (0.25)

Elle n'est pas en deuxième forme normale car il y a des attributs non-clé qui dépendent d'une partie de la clé. (0.25)

**4-** Normalisez la relation “Prime” jusqu’à la 3FN :

**Machine** (numtypeMachine, nomMachine) (0.5)

**Technicien** (numTechnicien, nomTechnicien) (0.5)

**Prime** (numTypeMachine, numTechnicien, montantPrime) (1)

**5-** indiquez si les relations obtenues sont en BCNF : Les trois relations sont en BCNF. (0.5)