

Examen du module: Système d'Information et Bases de Données - **Solution**

Exercice 1: (5 pts)

Q1 - Une base de données est une collection organisée de données connexes, alors que le SGBD est le logiciel qui permet la création, la gestion, et l'analyse de cette base de données.

(Définition BD : **0.5 pt**, Définition SGBD **0.5 pt**)

Q2 - Soit une partie de la base de donnée de gestion des cours d'une école définit par le diagramme ER comme suit :

a- Étudiant (id_étudiant, nom, prénom, date_naiss)

Étudiant_Tél (id_étudiant, numéro_téléphone)

Cours (id_cours, nom_cours, crédits, id_enseignant)

ÉtudCours (id_étudiant, id_cours)

Enseignant(id_enseignant, nom, prénom, date_naiss)

0.25 pt pour chaque clé primaire correcte

0.25 pt pour chaque table (nom , et structuration) à part pour la table ÉtudCours

0.5 pt pour la table ÉtudCours

0.25 pt pour le champ id_enseignant dans la table cours

-0.25 pt si l'attribut age est ajouté dans la table Enseignant

-0.25 pt pur chaque 2 autres attributs ajoutés

b- Non, on ne peut pas savoir si deux cours différents sont enseignés pendant le même créneau horaire **0.5 pt**.
Simplement parce que la donnée concernant l'attribution des créneaux horaires aux cours n'est pas disponible dans la modélisation décrite **0.5 pt**.

Exercice 2: (5 pts)

Q1 -

- ligne e_mail **1 pt**.

- pour les autres lignes,
chaque ligne **0.5 pt** ;

```
CREATE DATABASE LISTE;  
USE LISTE;  
CREATE TABLE Utilisateurs (  
  id INT UNIQUE NOT NULL,  
  nom char(50) NOT NULL,  
  e_mail char(50) CHECK (e_mail LIKE "%@%"),  
  CONSTRAINT id_pk PRIMARY KEY (id)  
);
```

Q2 -

- chaque ligne **0.5 pt.**

```
ALTER TABLE Utilisateurs  
ADD COLUMN prénom char(50);
```

Exercice 3: (6 pts)

Q1- a- la liste des employés (employé_id) ainsi que le salaire de chaque employé.

- chaque ligne **0.5 pt.**

```
SELECT employé_id, salaire  
FROM Employés;
```

b- la liste des différents départements (département_id) ainsi que le nombre d'employés dans chaque département unique.

- chaque ligne **0.5 pt.**

```
SELECT département_id, COUNT(employé_id)  
FROM Employés  
GROUP BY département_id;
```

c- la liste des employés (employé_id) ainsi que le nom du département auquel ils sont affectés.

- chaque ligne **0.5 pt.**

```
SELECT Employés.employé_id, Département.département_nom  
FROM Employés  
INNER JOIN Département On Employés.département_id = Département.département_id;
```

Q2-

- chaque ligne **0.5 pt.**

```
UPDATE Employés  
SET département_id = 3  
WHERE nom LIKE "n%";
```

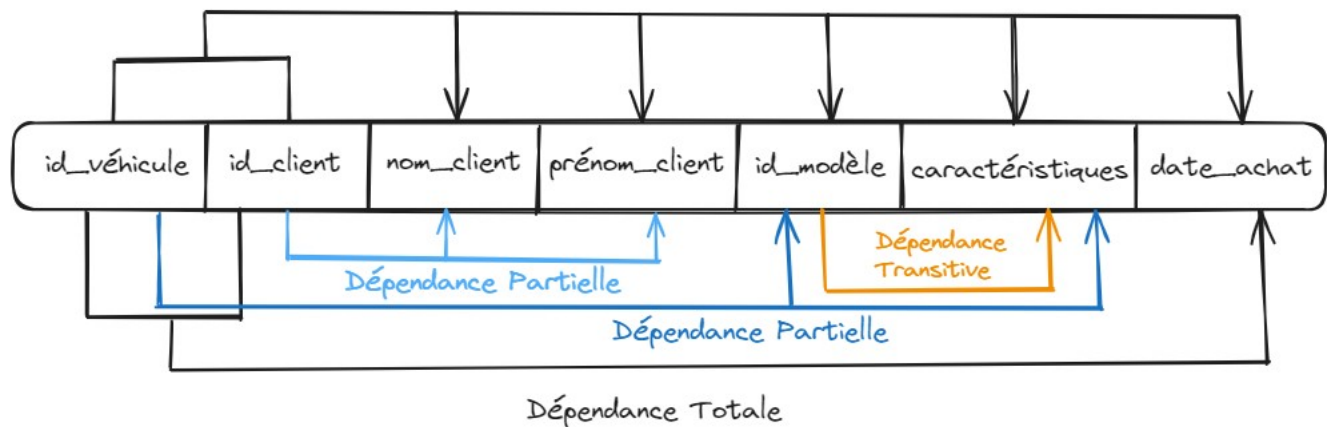
b- supprimer toutes les valeurs de la table « Employés»

- 0.5 pt.

```
DELETE FROM Employés;
```

Exercice 4: (4 pts)

Q1-



- 0.25 pt pour chaque dépendance partielle.

- 0.5 pt pour la dépendance totale

- 0.5 pt pour la dépendance transitive.

Q2-

- La relation est déjà en forme normale 1 (toutes les valeurs des attributs sont atomiques) 0.25 pt

- Forme Normale 2 :

Achat_Véhicule (id_véhicule, id_client, date_achat) 0.5 pt

Client (id_client, nom_client, prénom_client) 0.25 pt

Véhicule(id_véhicule, id_modèle, caractéristiques) 0.25 pt

- Forme Normale 3 :

Achat_Véhicule (id_véhicule, id_client, date_achat) - pas de changement -0.25 pt si changement est ajouté

Client (id_client, nom_client, prénom_client) - pas de changement -0.25 pt si changement est ajouté

Véhicule(id_véhicule, id_modèle) 0.25 pt

Modèles(id_modèle, caractéristiques) 0.5 pt