

Commissariat régional de l'éducation de Nabeul **** Devoir de synthèse N°3 Epreuve Théorique	Section : Sciences de l'Informatique	
	Matière : STI Systèmes & Technologies de l'Informatique	
	Niveau : 4ème année	
	date : 13/05/2025 durée : 3h Coefficient : 1.8	
Nom & Prénom :		Classe : Note : /20

Corrigé

Le sujet comportera 12 pages numérotées de "1/12" à "12/12".
Les réponses doivent être rédigées sur ces mêmes feuilles
qui doivent être remises à la fin de l'épreuve

EXERCICE N°1 (4 points = 8 * 0.5)

Pour chacune des affirmations ci-dessous, **une seule proposition** est **correcte**. Mettre une croix (X) dans la case correspondant à la proposition correcte.

Important : pour chaque question, toute réponse comportant plus d'une croix est considérée erronée

1) En HTML5, quel événement est déclenché lorsqu'une touche est enfoncée sur le clavier ?

- onkeyup
- onkeydown
- onmouseover

2) En HTML, quel attribut utiliser pour ouvrir la cible d'un lien dans une nouvelle fenêtre ?

- target="windows.open"
- target="new()"
- target="_blank"

3) En CSS3, quelle est la façon recommandée pour centrer un bloc horizontalement ?

- position: center;
- margin: auto;
- float : center ;

4) En CSS3, comment afficher une barre de défilement uniquement si le contenu d'un bloc déborde ?

- p {overflow: auto ; }
- p {overflow: hidden ; }
- p {overflow: visible ; }

5) En PHP, quel sera le résultat de ce code ?

- Erreur
- 3 + 6
- 9
- 3. + .6

```
<?php
    $a= 3;      $b =6;
    echo $a . "+" . $b;
?>
```

6) En PHP, la fonction die () ?

- a exactement le même rôle que la fonction echo
- arrête l'exécution d'un script PHP
- exécute une requête

7) En SQL, quelle sera la date retournée par la requête suivante ?

SELECT ADDDATE('2025-05-13', INTERVAL 7 DAY);

- '2025-05-19'
- '2025-05-20'
- '2025-12-13'

8) En SQL, quelle clause est utilisée pour filtrer les groupes en utilisant des fonctions d'agrégation telles que SUM(), COUNT()... ?

- GROUP BY
- ORDER BY
- HAVING

EXERCICE N°2 (8.75points)

L'énoncé figure dans la page 11

Répondre aux questions qui suivent en se référant à la représentation graphique de la base de données, aux règles de gestion et au tableau de description des champs de la page 11.

1) Identifier les clés primaires manquantes dans le tableau ci-dessous (0.5 pt)

Table	Clé primaire
Livreur	<i>Liv_id</i>
Commande	<i>Cde_id</i>
Client	Cli_id
Colis	<i>Colis_id</i>
Véhicule	Immatricule

2) Cette base de données contient une erreur liée au non-respect d'une règle de gestion.

a) Quelle est la règle de gestion qui n'a pas été respectée? Justifier votre réponse. (0.5pt)

La règle : R4

Justification : Absence de relation d'intégrité référentielle entre la table véhicule et la table commande

b) Pour corriger l'anomalie et respecter les règles de gestion, indiquez les liens entre les tables en plaçant une croix (X) dans chaque case où une table située en haut (table mère) est liée à une table située à gauche (table fille). (1.25 pt)

Table mère \ Table fille	Livreur	Commande	Client	Colis	Véhicule
Livreur					
Commande	X		X		X
Client					
Colis		X			
Véhicule	X				

c) Donner la nouvelle représentation textuelle des tables concernées pour corriger l'anomalie (0.5 pt)

Commande (NumCde, Montant, DateCde, Quantité, CodeLiv#, CodeCli#, Immatricule#)

Vehicule(Immatricule, Marque_V, Type_V, Liv_Id#)

3) Identifier pour chacune des propositions suivantes le type de la contrainte d'intégrité appropriée. (1.5 pts)

Proposition	Type de la contrainte
PRIMARY KEY (Liv_id)	Contrainte de table
Cli_id INT (4) REFERENCES Client (Cli_id)	Contrainte d'intégrité référentielle
Quantité INT (4) CHECK (Quantité >0)	Contrainte de domaine

4) Lors de la suppression d'un client, un message d'erreur apparaît.

Ecrire la requête SQL pour modifier la structure de la base de données afin de permettre la suppression des clients sans générer cette erreur ? **(0.75 pt)**

```
ALTER TABLE Commande DROP CONSTRAINT FK1;  
ALTER TABLE Commande ADD CONSTRAINT FK1 FOREIGN KEY (Cli_id)  
REFERENCES Client (Cli_id) ON DELETE CASCADE;
```

5) Lors de l'insertion des montants des commandes, l'administrateur a constaté une erreur : le champ "Montant" n'accepte pas la valeur 8575.575

Ecrire la requête SQL pour modifier la structure de la base de données et permettre la saisie de cette valeur ? **(0.5 pt)**

```
ALTER TABLE commande MODIFY Montant Decimal(7,3) DEFAULT 100.000;
```

6) Afin de distinguer les différents livreurs par leurs zones, on se propose de créer une nouvelle table nommée "Zone" ayant la description suivante :

Champ	Type et taille	Contrainte
Zone_id	Numérique	Clé primaire (incrémenté automatiquement)
Nom_Z	Chaîne de 15 caractères	Obligatoire
Code_Postal	Entier de 4 chiffres	Commence par 80 (Gouvernorat de Nabeul)

a) Écrire la requête SQL permettant de créer cette table. **(1.5pt)**

```
CREATE TABLE Zone (Code_Zone int (3) PRIMARY Key AUTO_INCREMENT,  
Nom_Z varchar(15) Not Null,  
Code_Postal int (4) not null CHECK (Code_Postal like '80%') );
```

b) On veut appliquer l'intégrité référentielle entre les tables "Livreur" et "Zone" en respectant les règles de gestion suivantes :

- R7 : Un livreur appartient à une seul Zone.
- R8 : Plusieurs livreurs peuvent appartenir à la même Zone. **(0.5 pt)**

Répondre par V si la clause proposée peut être utilisée pour relier les deux tables et par F sinon.

X	Zone_id INT (2) REFERENCES Zone (Zone_id)
	Zone_id INT (2) REFERENCES Livreur (Zone_id)
X	Zone INT (2) REFERENCES Zone (Zone_id)

7) Dans le cadre d'amélioration du système, l'administrateur a remarqué que le champ Adresse_Liv est un champ non déterminant dans la gestion de la base de données, il a décidé de le supprimer.

Écrire la requête SQL permettant d'effectuer cette tâche. **(0.25 pt)**

```
ALTER TABLE Livreur DROP Column Adresse_Liv ;
```

8) Dans ce qui suit, on suppose que la base de données est déjà créée et qu'elle contient des données :
Pour chacune des situations suivantes, mettre une croix (X) dans la (les) case(s) correspondante(s)
à la (aux) proposition(s) correcte(s). (1 pt)

a) En exécutant la requête « UPDATE Commande set Montant = Montant * 1.2 ; », le système :

- Augmente de 20% le Montant du **premier enregistrement** de la table
 Augmente de 20% le Montant de **tous les enregistrements** de la table
 Affiche un message d'erreur à cause de l'absence de la clause **WHERE**

b) L'administrateur veut supprimer la contrainte **DEFAULT** du champ montant de la table Commande.
Quelle requête lui convient ?

- ALTER TABLE Commande ALTER COLUMN Montant DROP DEFAULT ;
 ALTER TABLE Commande MODIFY COLUMN Montant DROP DEFAULT ;
 ALTER TABLE Commande MODIFY Montant DECIMAL (6, 3) ;

EXERCICE N°3 (27.25points)

 L'énoncé figure dans la page 12

PARTIE 1 : Programmation Web (Authentification) (10.75points)

A) En se référant à la représentation textuelle de la base de données, au tableau de description des champs et aux figures de la page 12, répondre aux questions qui suivent.

1) Compléter le **code html (ligne n°5)** par l'événement correspondant et la fonction « **Desactiver** » qui est déclenchée **au chargement de la page** et permettant de désactiver le bouton « **Connexion** » (1pt)

```
4 <script src="Controles.js"></script></head>
5 <body.....onload="Desactiver()" .....>
```

```
function desactiver () {
    bt1 = document.getElementById("bt1");
    bt1.disabled = true;
}
```

2) Compléter le **code html (ligne n°11)** et le script JavaScript (la fonction Colorer) pour colorer en jaune la zone de saisie (login) dès qu'elle prend **le focus**. (1pt)

```
10 <input name="T1" id="Log"placeholder="Login"type="text"class="input"
11 ...onfocus="colorer()".....>
```

```
function colorer () {
    Log = document.getElementById("Log");
    Log.style.backgroundColor = "yellow";
}
```

3) Pour des raisons de présentation, les caractères seront convertis en majuscule.

Compléter le code html (**ligne n°11**) et le script JavaScript (la fonction Convertir) pour convertir la chaîne login en majuscule dès que le champ **perd le focus** (*sans recopier la réponse de la question précédente*). (1pt)

```
10 <input name="T1" id="Log"placeholder="Login"type="text"class="input"
11 .....onblur="convertir()".....>
```

```
function convertir () {
  Log = document.getElementById("Log");
  Log.value = Log.value.toUpperCase(); }
}
```

4) Suite aux changements apportés sur les valeurs des zones login et mot de passe, une fonction JavaScript « **Verifier** » est appelée pour vérifier les conditions suivantes :

Champ	Contrainte
Login	Une chaîne de la forme XNNN où X est une lettre alphabétique et N un chiffre,
Mot de passe	Une chaîne de 4 chiffres.

- ❖ Si toutes les conditions de validation sont remplies, le bouton « **Connexion** » doit être **activé**.
- ❖ Si **au moins une** condition n'est pas remplie, un **message d'erreur** s'affiche en rouge **devant la zone de saisie concernée** (dans la zone output).
- ❖ Après correction d'une erreur, le message d'erreur est remplacé par le mot « **ok** », affiché en **vert**.
- ❖ Une fois **toutes** les erreurs **corrigées**, le bouton « **Connexion** » devient **actif**.

↳ Ci-après, on donne un script incomplet de la fonction « **Verifier** »

```
function Verifier(){
  var msg1 = document.getElementById("msg1");
  var msg2 = document.getElementById("msg2");
  var Ch1= document.getElementById("Log");
  var Ch2 = document.getElementById("MDP");
  var bt1 = document.getElementById("bt1");
  let a = false; let b = false;
  .....tache n°1 ..... 0.5
  if (.....tache n°2 .....) 1
  {msg1...tache n°3 ..... = " Erreur Login " ; 0.25
  msg1...tache n°4 .....= "red" ;} 0.25
  Else {
    msg1...tache n°3 ..... = "ok";
    msg1...tache n°4 .....= "green";
    a = true; }
  if (.....tache n°5 .....) 0.5
  {msg2...tache n°3 ..... = " Erreur Mot de passe " ;
  msg2...tache n°4 .....= "red" ;}
  Else {
    msg2...tache n°3 ..... = "ok";
    msg2...tache n°4 .....= "green";
    b = true; }
  bt1.disabled =.....tache n°6 .....; 0.5
}
```

Compléter chacune des tâches de la fonction « **Vérifier** » par la(les) condition(s) convenable(s) afin qu'elle réussisse à vérifier les conditions présentées par l'énoncé.

Tache	Condition(s) ou instruction(s) à écrire
Tache n°1	<code>var ch3 = ch1.substring(1, ch1.length);</code>
Tache n°2	<code>(ch1[0] > "Z" ch1[0] < "A" isNaN(ch3) ch3.indexOf(".") != -1 ch1.length != 4 ch3[0] == "-")</code>
Tache n°3	<code>.innerHTML</code>
Tache n°4	<code>.style.color</code>
Tache n°5	<code>(isNaN(ch2) ch2.indexOf(".") != -1 ch2.length != 4 ch2[0] == "-")</code>
Tache n°6	<code>!(a && b)</code>

5) En se référant au code HTML de la page "**index.html**", compléter le tableau ci-dessous soit par la description de la mise en forme soit par le code CSS adéquat. (3.25 pts) $4 \times 0.5 + 1.25$

Description de la mise en forme	Code CSS
L'arrière-plan de la page doit être de couleur 'GhostWhite'.	<code>body { background-color: GhostWhite; }</code>
Les boutons « Connexion et « Mot de passe oublié » doivent avoir une bordure de 2px , de type solid , et de couleur bleu .	<code>.bt { border: 2px solid blue; }</code>
Ajouter aux zones de saisie : <ul style="list-style-type: none"> • une marge extérieure inférieure de 10 pixels. • une marge intérieure de 10 pixels. 	<code>.input { margin-bottom: 10px; padding: 10px; }</code>
Les coins de la bordure des boutons "Connexion" et "Mot de passe oublié" sont arrondis de 10 pixels	<code>.bt { border-radius: 10px; }</code>
Ajouter un effet visuel au survol de l'image « logo.jpg » avec les effets suivants : <ul style="list-style-type: none"> • Agrandir le logo de 20 % • Le rendre légèrement transparent (opacité à 0.8) • Appliquer un flou de 2 pixels Utiliser la propriété transition pour animer ces effets en 0,3 seconde.	<code>#logo { transition-duration: 0.3s; } #logo:hover { transform: scale(1.2); opacity: 0.8; filter: blur(2px); }</code>

6) Compléter le fichier "**Connexion.php**" afin de permettre d'afficher le message "*Login ou mot de passe invalide*" dans le cas où une des valeurs saisies n'existe pas dans la base de données, sinon inclure la page "**Vérification.php**" (1.5 pt)

```
<?php
/* On suppose que la connexion au serveur et la sélection de la base sont déjà établies, */
/* On suppose que la variable de connexion est appelée $Con */
/* Instructions de récupération des données du formulaire F */
$Login = $_POST ['T1'];    $MDP = $_POST['T2'];
/* Instructions pour l'authentification du patient à partir des données du formulaire F */
$req="select * from Patient where MDP='$MDP' and IdP='$Login';
$res=mysqli_query($con,$req) or die ("echec dans la table Patient");
if (mysqli_num_rows($res)==0) {
    echo 'Login ou mot de passe invalide';
    exit;}
else {
    require "Vérification.php"
}
```

PARTIE 2 : Programmation Web (Vérification) (10.5pts)

On présente ci-dessous un extrait de la page "Vérification.php", qui reçoit les données du formulaire "F". Ces informations, **préalablement saisies** par le patient et transmises au laboratoire d'analyses, sont indispensables pour compléter le dossier médical et ainsi **interpréter de manière précise les résultats** des analyses effectuées.

Le **code du patient** ainsi que la **liste des analyses** qui lui sont associées sont récupérées via des **scripts PHP**.

Aperçu de la page "Vérification.php"

Code: Num Analyse:

Date du PRELEVEMENT :

Renseignements cliniques :

Pré-Op

Grossesse

Diabète

Chimiothérapie

Poids : 68

Les attributs name et id ont pour valeur : **NAn**

Les attributs name et id ont pour valeur : **Code**

Les attributs name et id ont pour valeur : **DP**

Les valeurs des attributs name et id sont respectivement : **R1, R2, R3 et R4**

La valeur de l'attribut id est **P**

Les attributs name et id ont pour valeur : **Pds**

- 1) Pour chacune des déclarations suivantes répondre par **V** si la déclaration **HTML5** proposée répond à la situation donnée, ou **F** dans le cas contraire, **(4 points)**

Important : toute case vide ou comportant une réponse autre que V ou F sera considérée erronée

- a) En **HTML**, pour créer la zone de saisie relative au champ de la date de prélèvement :

V	<code><input type="date" name="DP" id="DP" class="C1" ></code>
F	<code><input type="date-now" name="DP" id="DP" class="C1" ></code>
F	<code><input type="date-local" name="DP" id="DP" class="C1" ></code>

- b) En **HTML5**, pour créer la zone destinée au numéro d'analyse:

F	<code><selecte class="C1" id="NAn" name=" NAn"> <option >... </option></selecte></code>
V	<code><select class="C1" id=" NAn" name=" NAn"> <option >... </option></select></code>
F	<code><select class="C1" id=" NAn" name=" NAn" option="" ... > </select></code>

- c) En **HTML5**, pour créer la zone relative aux renseignements cliniques :

V	<code><fieldset><legend>Renseignements cliniques :</legend></code> <code><input type="checkbox" id="R1" name="R1" value="Pré-Op">Pré-Op</code> <code>...</fieldset></code>
F	<code><fieldset><legend>Renseignements cliniques :</legend></code> <code><input type="radio" id="R1" name="R1" value="Pré-Op">Pré-Op</code> <code>...</fieldset></code>
V	<code><fieldset></code> <code><input type="checkbox" id="R1" name="R1" value="Pré-Op">Pré-Op</code> <code>...<legend>Renseignements cliniques :</legend></fieldset></code>

d) En HTML5, pour créer la zone relative saisie du poids :

F	<input name="Pds" id="Pds" min="1" max="130" value="1" type="number">
V	<input name="Pds" id="Pds" min="1" max="130" value="1" type="range">
F	<input name="Pds" id="Pds" min="1" max="130" value="1" type="scrollbar">

e) En HTML5, pour créer la zone relative à l'affichage de la valeur du poids :

V	
V	<div id="P"></div>
V	<p id="P"></p>

f) En JavaScript, pour récupérer la valeur de la zone de saisie relative au champ de la date de prélèvement :

V	DP= document.getElementById("DP").value;
F	DP= document.getElementById("DP").datetime;
F	DP= document.getElementById("DP").date;

g) En JavaScript, pour récupérer la valeur de la zone relative au choix du numéro d'analyse :

F	index = document.getElementById('NAn').selectedvalue;
V	index = document.getElementById('NAn').selectedIndex;
V	selectElem = document.getElementById('NAn'); index = selectElem.selectedIndex;

h) En JavaScript, pour vérifier si la première case de la zone relative aux renseignements cliniques est choisie ou non :

V	If (document.getElementById("R1").checked==true);
V	R1=document.getElementById("R1").checked; If (R1==true);
F	R1=document.getElementById("R1").value; If (R1==true);

2) Lors du mouvement de la barre de progression (id="Pds"), la valeur du poids du patient est affichée **instantanément** (à chaque fois qu'une valeur est modifiée) dans la zone appropriée (id = "P").

On suppose que cette action est réalisée en faisant un appel à une fonction JavaScript nommée « Poids »

a) Quel est l'évènement HTML adéquat permettant de réaliser cette tâche ? (0.5 pt)

- onkeydown = "Poids()"
- onblur = "Poids()"
- oninput = "Poids()"

b) Compléter l'instruction manquante de la fonction « Poids » : (0.5 pt)

```
function Poids(){
    p=document.getElementById("Pds").value;
    document.getElementById("P").innerHTML = String(p);
}
```

3) Le clic sur le bouton « Valider » de la page « verification.php » déclenche l'exécution du script PHP « Valider.php », chargé de récupérer les informations du patient et de les mettre à jour.

Compléter le script « Valider.php » (le script doit afficher un message d'erreur si la mise à jour échoue ou si le script ne s'exécute pas, et un message de succès si la mise à jour est effectuée) (3pts)

```

<?php
    // On suppose que la connexion au serveur et la sélection de la base sont déjà établies et que la
    // variable de connexion est appelée $Con
    $NAn=   $_POST['NAn']; //Instruction de récupération du numéro d'analyse
    $DP=    $_POST ['DP']; // Instruction de récupération de la date du prélèvement
    $Pds=   $_POST['Pds']; // Instruction de récupération du poids du patient
    // Instructions de récupération des renseignements cliniques dans une chaîne de caractères $sch
    // et ils doivent être séparés par le caractère "|" afin de pouvoir les insérer dans la base de données.
    if ( isset ($_POST['R1']))
        {$sch= $sch. $_POST ['R1']. "|" ;}
    if ( isset ($_POST ['R2']))
        {$sch= $sch. $_POST ['R2']. "|" ;}
    if ( isset ($_POST ['R3']))
        {$sch= $sch. $_POST ['R3']. "|" ;}
    if ( isset ($_POST ['R4']))
        {$sch= $sch. $_POST ['R4']. "|" ;}
    $sch=substr ( $sch,0,strlen($sch)-1 );
    //Instruction(s) permettant la mise à jour des informations du patient (la table prescription)
    $req="update Analyse set DateAna='$DP', Poids ='$Pds', RensClin='$sch'
        where NumAna='$NAn' ";
    $res=mysqli_query($con,$req);
    if(mysqli_affected_rows($con)==0)
    {
        echo("Erreur d'exécution de la requête");}
    else {
        echo("Analyse modifiée avec succès"); }
?>

```

- 4) Compléter le script de la page "Afficher.php" ci-dessous permettant d'afficher les historiques des analyses d'un patient donné, on affichera **NomPrénomP** et **DateAna** (sans utiliser de tableau HTML) (2.5pts)

```

<?php
    // On suppose que la connexion au serveur et la sélection de la base sont déjà établies et que la
    // variable de connexion est appelée $Con
    $Code= $_POST [' Code ']; //instruction de récupération du code de patient.
    //Instruction(s) permettant l'affichage des analyses du patient.
    $req="select NomPrenomP, DateAna from Patient P, Analyse A, Ordonnance O
        where P.IdP=O.IdP and O.NumOrd=A.NumOrd and P.IdP='$Code' ";
    $res=mysqli_query($con,$req);
    while ($t=mysqli_fetch_array($res)) {
        echo $t['NomPrenomP']. " ". $t['DateAna']. "<br>";
    }
?>

```

PARTIE 3 : Gestion de la base de données (6pts = 1+ 0.75+1+1 + 0.75+ 0.75+ 0.75)

1) En se référant à la représentation textuelle et à la description des colonnes des tables figurant à la page 12, écrire les **requêtes SQL** permettant de répondre aux questions suivantes :

a) Afficher les informations des patients qui ont une analyse à la date courante **en ordre décroissant** selon leurs âges.

```
SELECT DISTINCT Patient.*
FROM Patient P, Analyse A, Ordonnance O
WHERE P.IdP = O.IdP and A.NumOrd = O.NumOrd
ORDER BY Pa.DateNaissP DESC;
```

b) Afficher **sans redondance** les spécialités de médecins enregistrés dans la base.

```
SELECT DISTINCT SpécialitéM
FROM Médecin
```

c) Afficher le code du médecin titulaire de la **dernière** ordonnance enregistrée dans la base.

```
SELECT CodeM
FROM Ordonnance
WHERE DateOrd = (SELECT MAX (DateOrd) FROM Ordonnance );
```

d) Afficher comme ci-contre le **nombre** d'ordonnances prescrites par chaque médecin.

NomPrenomM	Nbre_Ord

```
SELECT NomPrenomM , COUNT(O.CodeM) As Nbre_Ord
FROM medecin M, Ordonnance O
WHERE M.CodeM =O.CodeM
GROUP By NomPrenomM;
```

e) Supprimer toutes les analyses programmées durant le mois d'**avril 2025**

```
DELETE FROM Analyse WHERE Month (DateAna) =4 and Year (DateAna) =2025;
```

2) Donner le rôle de chacune des requêtes suivantes :

<p>a) SELECT M.NomPrenomM, Count(P.NumAna)As nb FROM Médecin M, Analyse A, Ordonnance O WHERE M.CodeM = O.CodeM AND A.NumOrd = O.NumOrd GROUP BY NomPrenomM HAVING nb > 10 ;</p>	<p>Chercher la liste des médecins qui ont au moins 10 analyses enregistrées <i>On affichera les noms suivis du nombre de prescriptions.</i></p>
<p>b) SELECT NomPrenomM FROM Médecin WHERE CodeM NOT IN (SELECT DISTINCT CodeM FROM Ordonnance)</p>	<p>Afficher les noms des médecins qui n'ont aucune ordonnance enregistrée.</p>

📄 Enoncé de l'exercice N°2

🚚 Société de services logistiques

La société **ADEXO** propose un service de livraison rapide et complet, disponible au quotidien. Elle regroupe et réexpédie des colis à domicile pour les entreprises comme pour les particuliers, en s'appuyant sur une large gamme de technologies innovantes dans le domaine du transport et de la logistique, afin de répondre efficacement aux besoins de ses clients.

Pour gérer les demandes des clients via l'application en ligne, le gérant utilise une base de données simplifiée nommée "**Gestion_livraison**", illustrée par une représentation graphique partielle, accompagnée de deux extraits de requêtes **SQL**.

Livreur	Commande	Client	Colis	Véhicule
Liv_id Nom_Liv Adresse_Liv Tel_Liv Statut	Cde_id Montant Date_Cde Quantité Liv_id Cli_id	Cli_id Nom_Cli Adresse_Cli Tel_Cli	Colis_id Prix Cde_id	Immatricule Marque_V Type_V Liv_id

```
CREATE TABLE Livreur (
Liv_id INT (8),
Nom_Liv VARCHAR (25) NOT NULL,
Adress_Liv VARCHAR (30) NOT NULL,
Tel_Liv VARCHAR (8) NOT NULL,
Statut VARCHAR (20) CHECK (Statut IN('actif', 'inactif')),
PRIMARY KEY (Liv_id));
```

```
CREATE TABLE Commande (
Cde_id int (6) PRIMARY KEY,
Montant DECIMAL (6, 3) DEFAULT 100.000,
Date_CDE DATE,
Quantité int (4) CHECK (Quantité >0),
Cli_id INT (4),
Liv_id INT (8) REFERENCES Livreur (Liv_id),
CONSTRAINT FK1 FOREIGN KEY (Cli_id)
REFERENCES Client (Cli_id) );
```

Soit la description des noms des colonnes des tables de cette base de données :

Nom	Description
Liv_id	Identifiant du livreur
Nom_Liv	Nom du livreur
Adresse_Liv	Adresse du livreur
Tel_Liv	Téléphone du livreur
Statut	Statut livreur ('actif' ou 'inactif')
Cde_id	identifiant de la commande
Montant	Montant de la commande
Date_Cde	Date de la commande
Quantité	Nombre de colis associés à la commande

Nom	Description
Cli_id	Identifiant du client
Nom_Cli	Nom du client
Adresse_Cli	Adresse du client
Tel_Cli	Téléphone du client
Colis_id	Identifiant de colis
Prix	Prix de colis
Immatricule	Immatricule de véhicule
Marque_V	Marque de véhicule
Type_V	Type de véhicule

Les règles de gestion sont :

- R1** : Un livreur peut livrer plusieurs commandes.
- R2** : Une commande est passée par un seul client.
- R3** : Un client peut passer plusieurs commandes.
- R4** : Une commande est transportée par un seul véhicule.
- R5** : Une commande peut contenir plusieurs colis.
- R6** : Un véhicule est conduit par un seul livreur.
- R7** : Un livreur peut conduire plusieurs véhicules

Enoncé de l'exercice N°3

Laboratoire d'analyses médicales

L'application « **LaboOnline** » permet aux patients de transmettre à l'avance certaines informations cliniques essentielles au laboratoire, afin de faciliter l'interprétation des résultats de leurs analyses médicales. La base de données permet de gérer les données liées aux patients, aux médecins, aux ordonnances, ainsi qu'aux prescriptions d'analyses.

⇒ Dans ce qui suit, nous allons nous intéresser exclusivement au formulaire via lequel le patient transmet ses informations cliniques nécessaires. Ce formulaire constitue un point d'entrée crucial dans le système, car il permet de recueillir des données utiles à l'analyse médicale avant même la réalisation des examens.

Ci-dessous, la représentation textuelle simplifiée de cette base de données.

Patient (IdP, NomPrenomP, GenreP, DateNaissP, TelP, MDP)

Médecin (CodeM, NomPrenomM, SpécialitéM, TelM)

Ordonnance (NumOrd, DateOrd, Description, **IdP#**, **CodeM#**)

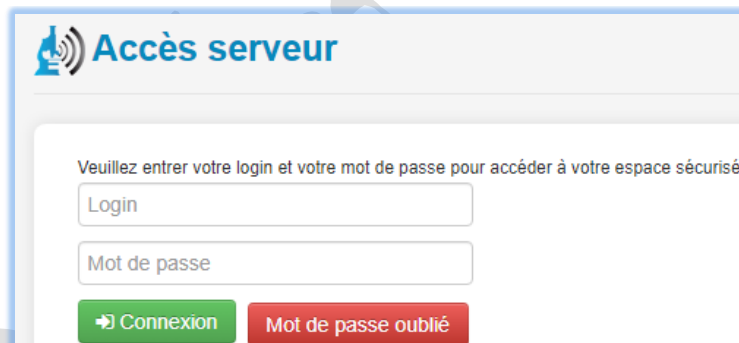
Analyse (NumAna, NumOrd#, DateAna, Poids, RensClin, Interprétation)

Soit la description des noms des colonnes des tables de cette base de données :

Nom	Description
MDP	Mot de passe du patient abonné
IdP	Identifiant du patient
NomPrenomP	Nom et prénom du patient
GenreP	Genre du patient
DateNaissP	Date de naissance du patient
TelP	Téléphone du patient
CodeM	Code du médecin
NomPrenomM	Nom et prénom du médecin
SpécialitéM	Spécialité du médecin

Nom	Description
TelM	Téléphone du médecin
NumOrd	Numéro de l'ordonnance
DateOrd	Date de l'ordonnance
Description	Détail de l'ordonnance
NumAna	Numéro de l'analyse
DateAna	Date de l'analyse
Poids	Mesure du poids du patient
RensClin	Informations cliniques du patient
Interprétation	Contenu de l'analyse

La figure ci-dessous représente la page "**index.html**" permettant à un patient (déjà inscrit(s)) d'entrer à l'espace du laboratoire via le formulaire suivant :



On donne ci-dessous un extrait du code HTML de la page "**index.html**".

```
1 <!DOCTYPE html>
2 <html><head><title>index</title>
3 <link type="text/css" rel="stylesheet" href="mestyles.css">
4 <script src="Controles.js"></script></head>
5 <body .....>
6 <header>Accès au serveur</header>
7 <fieldset name="fd1" >
8 <p>Veuillez entrer votre login et votre mot de passe pour accéder à votre espace sécurisé</p>
9 <form name="F" action="Connexion.php" method="post">
10 <input name="T1" id="Log"placeholder="Login" type="text" class="input"
11 .....><output id="msg1"></output>
12 <input name="T2" id="Mdp" placeholder="Mot de passe" type="password" class="input"
13 .....><output id="msg2"></output>
14 <input value="Connexion" type="submit" class="bt" id="bt1" >
15 <input value="Mot de passe oublié" type="reset" class="bt" id="bt2">
16 </form></body></html>
```