

L1 / IO2 Internet et Outils

Examen 16 Mai 2008 : 15H30-18H30

Document: 1 feuille A4 (recto-verso) autorisée

Total des points: 80. Votre note sera obtenue en divisant votre total de points par 4.

1. Réseau (Total: 20 points)

1.a. Décrivez brièvement les problèmes (au moins 4) qui se posent lorsque l'on veut permettre à des ordinateurs de communiquer entre eux. (4 points)

- *Quel moyen utiliser pour interconnecter les ordinateurs entre eux:*
 - *Connexion physique, médium, cable, ondes, diffusion ou point à point*
 - *Topologie des connexions*
 - *Impossibilité d'avoir des liaisons directes d'une machine vers toutes les autres, nécessité de passer par des machines intermédiaires pour acheminer des messages à destination.*
- *Acheminer les messages d'une machine à une autre*
- *Identifier les machines pour définir de manière non ambiguë la machine de destination et la machine émettrice.*
- *Hétérogénéité des machines (Big vs Little Endian), processeurs, OS*
- *Concurrence plusieurs machines communiquant avec la même*
- *Concurrence plusieurs applications sur une même machine accédant simultanément au réseau*

1.b. Qu'appelle-t-on "adresse IP" d'une machine? Quelles sont les caractéristiques d'une adresse IP? Donnez l'adresse IP valable sur chaque machine qui la désigne elle-même. Outre cette adresse, une machine peut-elle avoir plusieurs adresses IP? (4 points)

- *Identifiant d'une machine sur le réseau permettant de la distinguer des autres, en fait identifiant de l'interface de la machine et du réseau.*
- *Une adresse IP est supposée être unique sur le réseau Codée sur 32 bits. Représentation décimale pointée.*
- *Adresse valable sur chaque machine, la désignant elle même: 127.0.0.1*
- *Oui, il est possible d'attribuer plusieurs adresses IP pour une même machine, notamment pour celles qui sont connectées à plusieurs réseaux. Par ailleurs comme chaque machine est désignée par 127.0.0.1, la plupart des machines connectées ont au moins 2 adresses*

1.c. IPv6 va petit à petit être utilisé en remplacement de IPv4. Quel problème majeur IPv6 résout-il? Comment? (2 points)

- *Comme IPv4 code les adresses sur 32 bits, on va se trouver dans une situation d'épuisement des adresses IPv4 disponibles.*

- *IPV6 code les adresses sur 128 bits, et va donc éliminer ce problème d'épuisement des adresses disponibles.*

1.d. Pourquoi la désignation de machines par des adresses IP n'est-elle pas satisfaisante pour un utilisateur humain? Quelle solution a-t-on apporté à ce problème? Comment s'appelle le système de nommage mis en place pour résoudre ce problème? Décrivez la structure globale de ce système de nommage. Dans ce système de nommage, une machine peut-elle avoir plusieurs noms? (5 points)

- *Les adresses IP (surtout IPV6) ne sont pas simples à mémoriser, un nommage symbolique (basé sur des chaînes de caractères) est plus facile pour l'utilisateur humain*
- *On a donc inventé un système de nommage hiérarchique symbolique des machines connectées à Internet, appelé DNS (Domain Name System)*
- *L'administration du DNS est décentralisée*
- *On a défini des Top Level domains: com, gov, edu, org... et pays fr, us, uk*
- *L'administration de chaque domaine est confiée à un organisme indépendant (.fr => AFNIC) qui attribue les noms des sous_domaines à sa guise... et ainsi de suite.*
- *Les noms locaux peuvent ne pas être uniques. Les noms complets le sont forcément. Similaire à l'arborescence d'un système de fichiers.*
- *Une machine peut avoir plusieurs noms symboliques.*
- *Le DNS permet de convertir un nom symbolique (convivial pour un utilisateur humain) en une adresse IP.*

1.e. Citez deux commandes Unix permettant de transférer un ou plusieurs fichiers d'une machine à une autre. Comparez les. (4 points)

- *ftp*
- *scp: scp se rapproche de la commande cp (récuratif, travaille sur les arguments) ftp nécessite un sous-dialogue.*
- *Systèmes de fichiers distribués, P2P*
- *rsync: transfère seulement les différences entre une arborescence source et une distante.*

1.f. Citez au moins une commande Unix permettant d'exécuter un interpréteur de commandes interactif sur une machine distante (c-à-d autre que celle sur laquelle vous êtes connecté). (1 point)

- *telnet ou ssh*

2. MySQL (Total: 36 points)

Une association pour le développement du covoiturage en France, veut mettre en place un site web pour faciliter le partage de trajets. Chaque utilisateur devra s'enregistrer avant de pouvoir utiliser les services du site. Chaque propriétaire de véhicule pourra ensuite décrire ses trajets entre une ville de départ et une ville d'arrivée, en précisant la date du trajet et le nombre de places disponibles. Un passager pourra rechercher parmi les offres s'il existe un trajet correspondant à ses besoins.

Pour cela, le site web repose sur les 3 tables suivantes:

- Table des adhérents,
- Table des lieux,
- Table des trajets.

Chacune de ces tables est décrite ci-après:

- La table des adhérents contient les informations suivantes :
 - IdAdh : Identificateur numérique unique,
 - Nom : Chaîne de caractères de longueur variable d'au plus 30 caractères,
 - Prenom : Chaîne de caractères de longueur variable d'au plus 20 caractères,
 - Email : Chaîne de caractères de longueur variable d'au plus 48 caractères,
 - Tel : Chaîne de caractères de longueur variable d'au plus 15 caractères
 - Permis: Entier valant 1 si l'adhérent est titulaire du permis de conduire, 0 sinon.
 - Photo : Chemin d'accès à un fichier contenant une photo de l'adhérent. Ce champ ne sera utile que pour la dernière partie sur XHTML/PHP (Question 3.c)
- La table des lieux est définie comme suit :
 - IdLieu : Numéro unique attribué à chaque lieu,
 - Nom : Chaîne de caractères de longueur variable d'au plus 128 caractères,
 - Dept : Numéro du département dans lequel se trouve le lieu.
- La table des trajets est définie comme suit:
 - IdTraj : Numéro unique de trajet,
 - Conducteur : Identificateur du conducteur dans la table des adhérents,
 - Depart : Identificateur du lieu de départ dans la table des lieux,
 - Arrivee : Identificateur du lieu d'arrivée dans la table des lieux
 - Jour : de type "DATE": date prévue du voyage,
 - Passagers : Nombre maximum de passagers que peut accueillir le véhicule,
 - Libres : Nombre de places libres restant dans le véhicule.

Lors de son inscription, un adhérent doit obligatoirement fournir ses nom, prénom, adresse courriel. L'ensemble de ces trois informations doit être unique par adhérent, un adhérent est différent d'un autre si au moins une de ces trois informations diffère entre les 2 adhérents. Le numéro de téléphone et la photo ne sont pas obligatoires. Il est obligatoire d'indiquer si l'on est titulaire du permis de conduire. Le numéro unique d'adhérent est fourni automatiquement par le logiciel de gestion de la base de données MySQL.

Chaque lieu est identifié par une chaîne de caractères et le numéro du département. Ce couple d'informations identifie le lieu de manière unique. On peut donc avoir les 2 lieux suivants: Flagy (77 Seine et Marne) et Flagy (71 Saone et Loire). L'identificateur unique sera alloué automatiquement par le logiciel de gestion de la base de données MySQL.

Pour créer un trajet (une offre de trajet), un adhérent doit absolument fournir l'ensemble des informations listées dans la table "trajets". "Passagers" indique le nombre de passagers que le conducteur peut véhiculer. Le champ "Libres" est décrémenté à chaque fois qu'un passager s'inscrit pour ce trajet.

On manipulera les dates en utilisant des chaînes comme suit: 'AAAA-MM-JJ'. La date courante s'obtient en faisant appel à la fonction `CURDATE()`. Les dates peuvent se comparer avec les opérateurs de comparaison usuels: =, <, > etc...

Il est par ailleurs rappelé que l'on peut mettre à jour un champ d'un enregistrement en le référençant. Par exemple, on peut diminuer le nombre de places disponibles pour un trajet en écrivant dans la commande de mise à jour: `Libres=Libres - 1`.

2.a. Écrivez la requête MySQL nécessaire pour créer la table des Adhérents. Veillez à ce que les contraintes décrites ci-dessus soient vérifiées par la base de données lors des requêtes d'insertion dans la table. (4 points)

```
CREATE TABLE Adherents (
  -> IdAdh INTEGER NOT NULL AUTO_INCREMENT,
  -> Nom VARCHAR(30) NOT NULL,
  -> Prenom VARCHAR(20) NOT NULL,
  -> Email VARCHAR(48) NOT NULL,
  -> Tel VARCHAR(15),
  -> Permis INTEGER DEFAULT 0,
  -> PRIMARY KEY(IdAdh),
  -> UNIQUE (Nom, Prenom, Email)
  -> );
```

2.b. Écrivez la requête MySQL nécessaire pour créer la table des lieux. Veillez à ce que les contraintes décrites ci-dessus soient vérifiées par la base de données lors des requêtes d'insertion dans la table. (2 points)

```
CREATE TABLE Lieux (
  -> IdLieu INTEGER AUTO_INCREMENT,
  -> Nom VARCHAR(128) NOT NULL,
  -> Dept INTEGER NOT NULL DEFAULT 75,
  -> PRIMARY KEY (IdLieu),
  -> UNIQUE (Nom, Dept)
  -> );
```

2.c. Écrivez la requête MySQL nécessaire pour créer la table des trajets. Veillez à ce que les contraintes décrites ci-dessus soient vérifiées par la base de données lors des requêtes d'insertion dans la table. (2 points)

```
CREATE TABLE Trajets (
  -> IdTraj INTEGER NOT NULL AUTO_INCREMENT,
  -> Conducteur INTEGER NOT NULL,
```

-> *Depart INTEGER NOT NULL,*
 -> *Arrivee INTEGER NOT NULL,*
 -> *Date DATE NOT NULL,*
 -> *Passagers INTEGER NOT NULL DEFAULT 3,*
 -> *Libres INTEGER DEFAULT 3,*
 -> *PRIMARY KEY (IdTraj),*
 -> *UNIQUE(Conducteur, Depart, Arrivee, Date),*
 -> *FOREIGN KEY(Conducteur) REFERENCES Adherents,*
 -> *FOREIGN KEY(Depart) REFERENCES Lieux,*
 -> *FOREIGN KEY(Arrivee) REFERENCES Lieux*
 -> *);*

2.d. On suppose que la table des adhérents est initialement vide. Écrivez la requête MySQL nécessaire pour enregistrer le premier adhérent suivant: (2 points)

- André Citroën, ac@2CV.org, Tel: +336 1234 5678, Permis B 12 points.

INSERT INTO Adherents (Nom, Prenom, Email, Tel, Permis)

-> VALUES ("Citroen", "Andre", "ac@2cv.org", "+336 1234 5678", 1);

2.e. Expliquez s'il est ensuite possible d'enregistrer les personnes ci-dessous dans la table des adhérents après l'enregistrement du premier adhérent. Motivez vos réponses. (3 points)

- Albert De Dion, Tel: +332 1234 5678, Pas de permis.

Non, l'information 'email' indispensable n'est pas disponible.

- Xantia Citroën, ac@2CV.org, Ni téléphone, ni permis.

Oui, le triplet (nom, prénom, email) est différent du triplet pour André Citroen déjà enregistré.

- André Citroën, andre.citroen@paradis.org, Permis B 6 points, pas de téléphone.

Oui, le triplet (nom, prénom, email) est différent du triplet pour André Citroen déjà enregistré.

Par la suite, on suppose que la table des lieux contient déjà les lieux suivants:

Nom	Dept
Paris	75
Versailles	78
Flagy	77
Flagy	71

2.f. Écrivez la(les) requête(s) MySQL nécessaire(s) pour enregistrer le fait que l'adhérent

André Citroën propose un trajet de Paris à Lyon (69) le 1 Juin 2008, pour 3 passagers.
(4 points)

INSERT INTO Lieux (Nom, Dept) VALUES ("Lyon", 69);

INSERT INTO Trajets (Conducteur, Depart, Arrivee, Date)

-> VALUES (

(SELECT IdAdh From Adherents WHERE Nom="Citroen" AND Prenom="Andre"),

(SELECT IdLieu From Lieux WHERE Nom="PARIS"),

(SELECT IdLieu From Lieux WHERE Nom="LYON" AND Dept=69),

"2008-06-01");

On pouvait aussi indiquer que l'on supposait les valeurs IdAdh et IdLieu nécessaires connus.

2.g. On suppose que l'adhérent Gérard Levasseur gl@panhard.com, préalablement enregistré dans la base de l'association, ne souhaite plus faire partie de l'association et n'est partie prenante dans aucun trajet que ce soit en tant que conducteur ou en tant que passager. Écrivez la requête MySQL nécessaire pour supprimer l'enregistrement de cet adhérent. (2 points)

DELETE FROM Adherents WHERE Nom="Levasseur" AND Prenom="Gerard" ;

2.h. Écrivez la requête MySQL nécessaire pour lister l'ensemble des adhérents de l'association enregistrés classés par ordre alphabétique croissant de nom de famille, et par ordre alphabétique décroissant de prénom pour les adhérents ayant le même nom de famille. (2 points)

SELECT Nom, Prenom FROM Adherents ORDER BY Nom, Prenom DESC;

2.i. Écrivez la requête MySQL nécessaire pour mettre à jour les informations du trajet prévu par André Citroën de Flagy(77) à Lyon le 13 Juin 2008 pour 3 passagers pour que ce trajet soit maintenant annoncé pour le 12 Juin 2008 et pour 2 passagers. On supposera qu'il n'y avait pas de passager ayant déjà sélectionné ce trajet. (3 points)

UPDATE Trajets SET Date="2008-06-13", Passagers=2, Libres=2 WHERE IdTraj=3;

En précisant comment IdTraj est obtenu, ou mieux:

UPDATE Trajets SET Date="2008-06-13", Passagers=2, Libres=2

WHERE

Conducteur=

(SELECT IdAdh FROM Adherents WHERE Nom="Citroen" AND Prenom="Andre")

AND

Depart=(SELECT IdLieu From Lieux WHERE Nom="FLAGY" AND Dept=77)

AND

Arrivee=(SELECT IdLieu From Lieux WHERE Nom="LYON" AND Dept=69)

AND

Jour="2008-06-12";

2.j. Écrivez la requête MySQL permettant d'afficher les trajets à venir au départ de Paris

pour lesquels il reste des places disponibles. On affichera de manière non ambiguë la ville de destination, les nom, prénom, email du conducteur, la date du voyage et le nombre de places encore disponibles. (4 points)

```
SELECT Lieux.Nom, Adherents.Nom, Adherents.Prenom, Adherents.Email,
       Trajets.Date, Libres FROM Trajets, Lieux, Adherents
-> WHERE Depart= (SELECT IdLieu FROM Lieux Where Nom ="Paris")
-> AND Trajets.Arrivee = Lieux.IdLieu
-> AND Trajets.Conducteur = Adherents.idAdh
-> AND Libres != 0
-> AND Date > CURDATE();
```

2.k. Écrivez la requête MySQL permettant de supprimer de la base tous les trajets dont la date est antérieure à la date du jour. (2 points)

```
DELETE FROM Trajets WHERE Date < CURDATE();
```

2.l. Quelle est la requête MySQL qu'un adhérent désirant réserver une place sur un trajet de Versailles à Flagy (71) à la date du 11 Juillet 2008 dans le véhicule de Enzo Ferrari (ef@ferrari.com) doit effectuer? (2 points)

```
UPDATE Trajets SET Libres=Libres-1
```

```
WHERE
```

```
Conducteur=
```

```
(SELECT IdAdh FROM Adherents WHERE Nom="Ferrari" AND Prenom="Enzo")
```

```
AND
```

```
Depart=(SELECT IdLieu From Lieux WHERE Nom="VERSAILLES" AND Dept=78)
```

```
AND
```

```
Arrivee=(SELECT IdLieu From Lieux WHERE Nom="FLAGY" AND Dept=71)
```

```
AND
```

```
Jour="2008-06-11";
```

2.m. Peut-on avec les tables de la base telles qu'elles sont définies, savoir si certains des passagers pourront éventuellement relayer le conducteur pour la conduite? Si oui, donnez la requête MySQL correspondante. Si non, expliquez pourquoi et décrivez quelle solution vous envisageriez. (4 points)

Non. Il n'y a pas de possibilité de retrouver les passagers enregistrés pour un trajets? Seul le conducteur proposant le trajet est connu.

Pour pouvoir savoir si des passagers pourraient participer à la conduite, il faudrait au choix:

- *Ajouter la référence des passagers dans la table des trajets, mais on serait obligé d'imposer une limite sur le nombre de passagers,*
- *Ajouter une référence aux trajets dans la table des Adhérents, mais on imposerait une limite sur le nombre de trajets auxquels peut participer à un momemnt donné,*
- *Ou ajouter une table faisant référence aux trajets et aux adhérents.*

N'importe quelle des 3 solutions ci-dessus est une bonne réponse.

3. PHP / XHTML (Total: 24 points)

On veut obtenir le formulaire suivant pour enregistrer les adhérents de l'association. Il est possible pour l'adhérent de fournir une photo qui sera transférée de la machine client vers le serveur ("upload").

Les champs écrits en **rouge** doivent être obligatoirement renseignés.

Nom	<input type="text"/>	Prénom	<input type="text"/>
E-mail	<input type="text"/>	Téléphone	<input type="text"/>
Permis de Conduire?	OUI <input type="radio"/> NON <input checked="" type="radio"/>		
Votre photo:	<input type="text"/>		<input type="button" value="Browse..."/>
<input type="button" value="Envoi!"/>		<input type="button" value="On recommence tout"/>	

La base a pour nom: "COVOITURAGE", elle est accessible par un utilisateur "ADMIN" avec le mot de passe "PASSEPASSE".

3.a. Écrivez l'extrait de code XHTML nécessaire pour afficher ce formulaire. On utilisera un tableau et des clauses CSS pour aligner les noms des champs et indiquer en rouge ceux qui doivent être obligatoirement renseignés. On ne vous demande pas le code complet d'un document XHTML, mais seulement l'extrait nécessaire à la réalisation de l'affichage demandé. (6 points)

`<html>`

`<head>`

`<title>Exemple simple de formulaire</title>`

`<style type="text/css">`

`td.droit {text-align:right}`

`*.oblig {color:red} /* Obligatoire */`

`*.centre {text-align:center}`

`</style>`

`</head>`

`<body style="background-color: white">`

`<h1 class="centre"> Un formulaire</h1>`

`<p>Les champs écrits en rouge`

`doivent être obligatoirement renseignés.`

`<form enctype="multipart/form-data" action="covoiture.php" method="post">`

`<table border="4">`

`<tr>`

`<td class="droit oblig">Nom</td>`

`<td><input type="text" name="nom" size="30" maxlength="60"/> </td>`

```

<td class="droit oblig">Pr&eacute;nom</td>
  <td><input type="text" name="prenom" size="30" maxlength="60" /> </td>
</tr>
<tr>
  <td class="oblig">E-mail</td>
  <td><input type="text" name="email" size="30" maxlength="60" /> </td>
  <td>T&eacute;l&eacute;phone</td>
  <td><input type="text" name="Tel" size="30"/> </td>
</tr>
<tr>
  <td class="oblig">Permis de Conduire? </td>
  <td colspan="3">
    OUI<input type="radio" name="permis" value="oui" />
    NON<input type="radio" name="permis" value="non" />
  </td>
</tr>
<tr class="centre">
  <td> Votre photo:</td>
  <td class="centre" colspan=3>
    <input type=hidden name="MAX_FILE_SIZE" value="3000000">
    <input type="file" size="80" name="photo">
  </td>
</tr>
<tr>
  <td colspan="2" style="text-align:center">
    <input type="submit" value="Envoi!" name="go" />
  </td>
  <td colspan="2" style="text-align:center">
    <input type="reset" value="On recommence tout" />
  </td>
</tr>
</table>
</form>
</body>
</html>

```

3.b. Quelle URL sera envoyée par la navigateur quand l'adhérent Emile Fourtoux

(ef@ef.org, +331 0987 6543, sans permis de conduire, sans photo) aura rempli ce formulaire et cliqué sur le bouton Envoi? (4 points)

Si on a utilisé la méthode POST: localhost://covoiture.php

Si on a utilisé la méthode GET:

localhost://covoiture.php?nom=Fourtoux&prenom=Emile&email=ef@ef.org&permis=0&33109876543&envoi=go

Des formes moins complètes de réponses étaient acceptables.

3.c. Écrivez le code du script PHP qui permet de traiter correctement le formulaire ci-dessus en vérifiant que les champs obligatoires sont bien remplis et en renvoyant les messages d'erreur appropriés dans le cas contraire. On traitera correctement la récupération de la photographie, et on procédera à l'enregistrement de l'utilisateur dans la base de données de l'association. (14 points)

Il est vivement conseillé de structurer le code PHP.