

Examen National de Fin d'année
Session de Septembre 2020

Examen de Fin de Formation (Epreuve de Synthèse)

<u>Filière</u>	Techniques de Développement Informatique	<u>Variante</u>	Principale
<u>Niveau</u>	TS	<u>Durée</u>	4 H
		<u>Barème</u>	/100

Consignes et Conseils aux candidats :

IMPORTANT

- Les dossiers 1, 2 et 3 sont OBLIGATOIRES.
- Vous avez le C H O I X entre le dossier 4 ou le dossier 5.

Document(s) et Matériel(s) autorisés :

Aucun document ou matériel n'est autorisé.

Détail du Barème :

N° Des Dossiers	Travaux à réaliser	Barème
Partie Théorique		
Dossier 1	Base de données (les bases)	30 points
Dossier 2	Programmation Orientée Objet	10 points
		/40 points
Partie Pratique		
Dossier 3	Base de données (requête /procédures/Triggers)	30 points
Dossier 4	Développement Web coté client (au choix)	30 points
Dossier 5	Développement client-serveur (au choix)	30 points
		/60 points
Total Général		/100 points

Dans le cadre des activités du festival « Marrakech du Rire », le comité d'organisation veut lancer une compétition pour choisir la meilleure blague qui sera présentée sur scène. Pour cela on veut concevoir et développer une plateforme permettant aux internautes de s'inscrire pour participer avec des blagues ou de voter pour une blague en tant que visiteur seulement. On a la base de données *BD_Festival* suivante :

- > **Ville** (idV, libelleV)
- > **Participant** (idP, nom, prenom, civilite, tel, idV#)
- > **Blague** (idBlague, idP#, intitule, contenu, nombreVote)
- > **Vote** (IDSESSION, idBlague#, date_vote)
 - Les champs en **gras et soulignés** sont des clés **primaires**, les champs suivis par # sont des clés **étrangères**.
 - Le champ **IDSESSION** représente l'identifiant de la session de l'internaute.

- 1) Qu'il est le rôle d'une clé primaire d'une table. **(4pts)**
- 2) Qu'il est le rôle d'une clé étrangère d'une table. **(4pts)**
- 3) Donner toutes les contraintes d'intégrité référentielle qui se trouvent dans ce schéma de base de données. **(4pts)**
- 4) Peut-on supprimer une ligne de la table participant qui a des lignes correspondantes dans la table Blague? justifier votre réponse. **(4pts)**
- 5) Selon ce modèle, un visiteur peut-il voter pour plusieurs blagues? justifier votre réponse. **(4pts)**
- 6) un participant peut-il participer avec plusieurs blagues? justifier votre réponse. **(4pts)**
- 7) Proposer un modèle conceptuel de données (MCD) correspondant au schéma relationnel ci-dessus. **(6 pts)**

Dossier 2 : (10 pts)

On veut implémenter une classe qui représente un participant au festival « Marrakech du Rire ». On retient pour un participant les champs : idP, nom, prenom et tel.

- 1) Ecrire la classe Participant. **(3pts)**
- 2) Ajouter le constructeur par défaut et un constructeur d'initialisation. On doit s'assurer que le N° de téléphone du participant soit composé de 10 chiffres en commençant par 0. **(3pts)**
- 3) Ajouter les accesseurs de la classe Participant. **(2pts)**
- 4) Ecrire une méthode qui retourne une chaîne de caractères donnant le nom complet d'un participant sous la forme : « P.NOM » (1^{er} caractère du prénom en majuscule concaténé avec le nom et séparé par le caractère point « . »). **(2pts)**

On reprend la base de données du dossier 1 :

- > **Ville** (idV, libelleV)
- > **Participant** (idP, nom, prenom, civilite, tel, idV#)
- > **Blague** (idBlague, idP#, intitule, contenu, nombreVote)
- > **Vote** (IDSESSION, idBlague#, date_vote)

- 1) Ecrire une requête pour supprimer toutes les blagues ayant moins de 3 votes. **(5pts)**
- 2) Ecrire une requête qui affiche le nom et prénom de tous les participants de la ville de « Fes ». **(5pts)**
- 3) Ecrire une requête qui affiche le nombre de blagues par participant. On affiche le nom et prénom ainsi que le nombre de blagues. On retient seulement ceux ayant plus de 2 blagues. **(5pts)**
- 4) Ecrire une requête qui affiche le nom, prénom et le nombre de vote pour chaque blague des trois participants ayant obtenu les trois premières places dans le classement. N'utiliser pas le champ *nombrevote*, recalculer le. **(5pts)**
- 5) Ecrire une procédure stockée permettant de renvoyer à l'aide d'un paramètre de sortie le nombre de vote obtenu par un participant dont l'id est passé en paramètre. **(5pts)**
- 6) Ecrire un trigger permettant d'incrémenter le champ *nombrevote* de la table Blague suite à chaque nouveau vote. **(5pts)**

Traiter au choix le dossier 4 ou le dossier 5 :

Dossier 4 : (30 pts)

On veut publier sur le web la base de données du festival des blagues présenté dans le dossier 1. Pour cela on a exporté les données de la base de données dans le fichier XML suivant :

```
<?xml version="1.0" ?>
<FestivaleRire>
  <participant idp="1">
    <nom>nom1</nom>
    <prenom>prenom1</prenom>
    <civilite>Mr</civilite>
    <tel>000000000</tel>
    <ville>
      <idv>1</idv>
      <libellev>Rabat</libellev>
    </ville>
    <blague idb="1">
      <titre>titre1</titre>
      <contenu>contenu1</contenu>
      <nbrvote>2</nbrvote>
```

Filière	Techniques de Développement Informatique	Variante		Page	Page 3 sur 6
Examen	Fin de Formation / Synthèse	Nbr de page	6		

```

</blague>
</participant>
<participant idp="2">
  <nom>nom2</nom>
  <prenom>prenom2</prenom>
  <civilite>Mr</civilite>
  <tel>000000000</tel>
  <ville>
    <idv>1</idv>
    <libellev>Fes</libellev>
  </ville>
  <blague idb="2">
    <titre>titre2</titre>
    <contenu>contenu2</contenu>
    <nbrvote>3</nbrvote>
  </blague>
</participant>
...
</FestivaleRire>

```

- 1) Proposer un schéma XSD pour représenter les données de ce fichier XML. (5pts)
- 2) Ecrire une expression XPath qui sélectionne les blagues du participant d'identifiant (idp) égal à 2. (5pts)
- 3) Ecrire une expression XPath qui compte le nombre de participant de la ville Rabat. (5pts)
- 4) On veut permettre à un internaute de s'inscrire afin de participer avec une blague. Ecrire le code HTML du formulaire d'inscription suivant : (5pts)

Inscription:

Madame
 Mademoiselle
 Monsieur

Votre prenom :*

Votre nom :*

Telephone :*

Ville *

Intituler de votre blague :*

Votre Blague * :

Quel est le resultat de l'expression : 3+8=

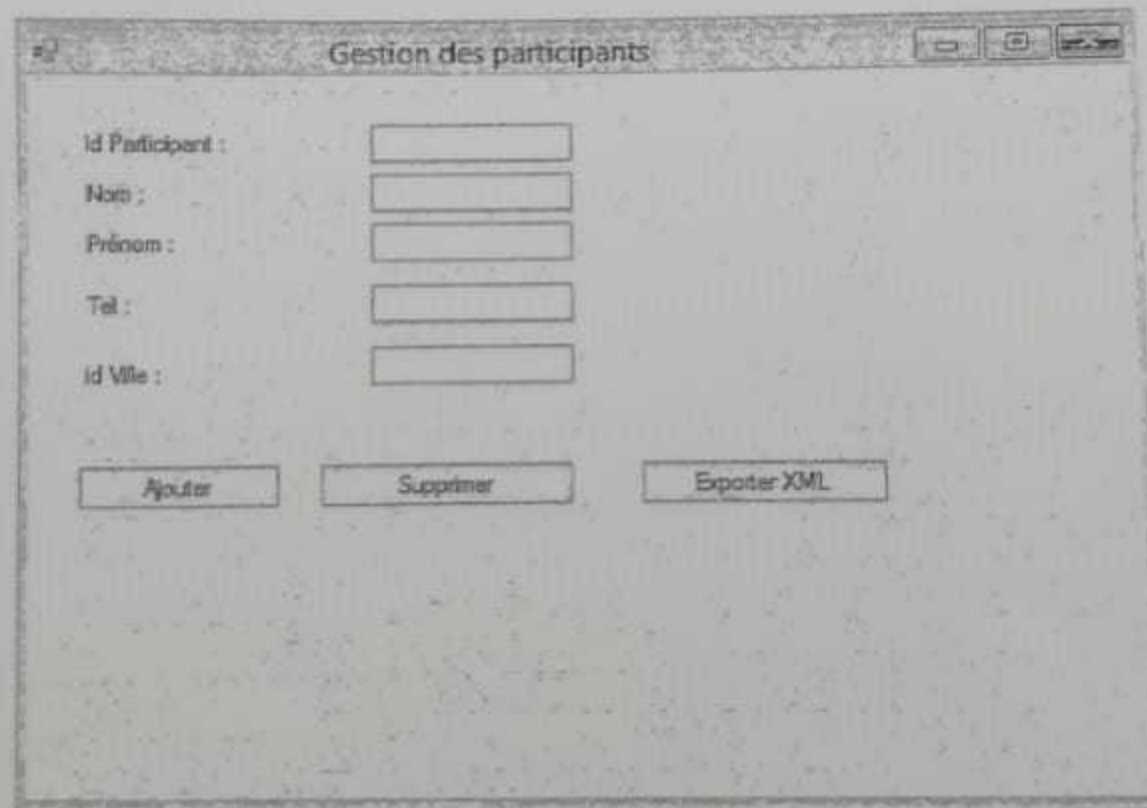
- a. Tous les champs sont obligatoires.
- b. Valider le type de donnée saisie pour chaque colonne.

- 5) On veut au clic sur le bouton Sauvegarder, valider l'identité de l'internaute pour s'assurer qu'il ne s'agit pas d'un robot. On insère dans le formulaire une expression générée aléatoirement dont l'utilisateur doit fournir la réponse. Utiliser la fonction JavaScript `Math.random()` permettant de générer des valeurs entre 0 et 1 (0 inclusive, 1 exclusive). (5pts)
- 6) Ecrire le code JavaScript permettant suite au clic sur le bouton sauvegarder, d'afficher tous les champs saisi dans le formulaire. (5pts)

Dossier 5 : (30pts)

Une application client-serveur utilise la base de données *BD_Festival* du dossier 1.

On a le formulaire suivant qui permet de gérer les participants :



The screenshot shows a web browser window with the title "Gestion des participants". Inside the window, there is a form with five input fields, each with a label to its left: "Id Participant", "Nom", "Prénom", "Tel", and "id Ville". Below the input fields, there are three buttons: "Ajouter", "Supprimer", and "Exporter XML".

- 1) Ecrire le code des boutons « Ajouter » et « Supprimer » qui permettent respectivement d'ajouter et de supprimer un participant. (5pts)
- 2) Ecrire le code du bouton « Exporter XML » qui exporte tout le contenu de la table Participant vers un fichier de format XML. (5pts)
- 3) Ecrire la méthode `void AjouterBlague (int idBlague , int idP , string intitule , string contenu)` qui permet d'ajouter une ligne à la table Blague : (5pts)

• Les paramètres de la méthode correspondent aux colonnes de la table Blague ;

• La méthode doit vérifier que le participant de code **idP** existe dans la table **Participant** ; sinon elle affiche le message d'erreur « Participant inexistant ».

- 4) Écrire la méthode **void SupprimerBlagues (int idP)** qui permet de supprimer toutes les blagues proposées par le participant de code **idP**. (5pts)
- 5) Écrire la méthode **int Votes (DateTime DateDebut, DateTime DateFin)** qui retourne le nombre de votes effectués entre deux dates données (paramètres de la méthode). (5pts)
- 6) Écrire la méthode **ArrayList lesBlagues ()** qui retourne une collection (de type **ArrayList** ou **List**) contenant le code (**idBlague**) des blagues qui ont obtenu les 3 premières places en nombre de votes. (5pts)