

Exercice N°1

*Cet exercice porte sur des connaissances générales dans un contexte
de bases de données et de développement Web*

Pour chacune des propositions ci-dessous, répondre par la lettre (V) si la proposition est correcte ou par la lettre (F) dans le cas contraire.

1. En PHP, le résultat de l'exécution d'une requête de sélection (SELECT) est :

- ☐ une chaîne
- ☐ un formulaire
- ☐ un fichier
- ☐ un tableau

2. En SQL, il est possible d'insérer une sous requête dans une requête :

- ☐ en utilisant l'opérateur "="
- ☐ après la clause **WHERE**
- ☐ en utilisant la clause **IN**
- ☐ en utilisant la clause **BETWEEN**

3. En PHP, la déclaration d'une variable peut prendre la forme :

- ☐ \$ma15variable ;
- ☐ \$ma_15variable = 1 ;
- ☐ \$_mavariab15 ;
- ☐ \$15mavariab1 ;

4. En SQL, dans une requête de création de table, on peut utiliser le mot clé :

- ☐ ORDER BY
- ☐ DISTINCT
- ☐ GROUP BY
- ☐ UNIQUE

5. En SQL, la clause HAVING :

- ☐ doit être utilisée obligatoirement avec la clause GROUP BY
- ☐ doit être utilisée avec une fonction agrégat
- ☐ est similaire à une clause WHERE, mais, au lieu de filtrer des lignes, elle filtre des groupes
- ☐ est équivalente à la clause WHERE

Exercice N°2

Cet exercice porte sur le développement Web

Pour chaque proposition de la colonne **Bloc**, compléter la colonne **Réponse** par la lettre qui correspond au **Langage** approprié.

| Bloc | | Réponse | Langage | |
|------|----------------------------------|---------|---------|------------|
| 1 | for (\$i=0 ; \$i<\$N ; \$i++) | 1 | a | JAVASCRIPT |
| 2 | Alert (" Texte "); | 2 | b | PHP |
| 3 | \$req="SELECT * From Matiere " ; | 3 | c | HTML |
| 4 | for (i=0 ; i<N ; i++) | 4 | d | SQL |

Exercice N°3

Cet exercice porte sur la gestion de bases de données et le développement Web

La société "**NetFilm**" propose des services de streaming pour regarder des films en ligne en offrant un abonnement à chaque famille. Elle propose également un service de votes permettant à un membre d'une famille d'évaluer les films en attribuant un nombre d'étoiles exprimant son appréciation.

Ci-dessous, la représentation textuelle simplifiée de la base de données conçue pour gérer les abonnés et les services de streaming.

Famille (IdFamille, NomFamille, EmailFamille, TelephoneFamille)

MembreFamille (IdMembre, Pseudo, IdFamille#)

Realisateur (IdRealisateur, NomRealisateur)

Film (IdFilm, Titre, Description, IdRealisateur#, Duree)

Voter (IdFilm#, IdMembre#, NombreEtoiles)

Description des champs des tables

| Champ | Description |
|------------------|---------------------------------------|
| IdFamille | Identifiant d'une famille |
| NomFamille | Nom d'une famille |
| EmailFamille | Email d'une famille |
| TelephoneFamille | Numéro de téléphone d'une famille |
| IdMembre | Identifiant d'un membre d'une famille |
| Pseudo | Pseudonyme du membre d'une famille |
| IdRealisateur | Identifiant d'un réalisateur |

| Champ | Description |
|----------------|--------------------------|
| NomRealisateur | Nom d'un réalisateur |
| IdFilm | Identifiant d'un film |
| Titre | Titre d'un film |
| Description | Description d'un film |
| Duree | Durée d'un film |
| NombreEtoiles | Note attribuée à un film |

Partie A : Gestion de données

1. Ecrire la requête SQL permettant d'afficher :
 - a. l'identifiant et le pseudo de tous les membres de famille d'identifiant 528.
 - b. la liste des films (Titre, Description) réalisés par "ChawkiMejri".
 - c. pour chaque film, le titre et la moyenne des notes attribuées.
 - d. la liste des films (IdFilm, Titre) ayant la plus longue durée.
2. Donner le rôle de chacune des requêtes suivantes :
 - a. UPDATE Voter
SET NombreEtoiles = NombreEtoiles + 1
WHERE IdFilm = 'A1057' AND IdMembre = 400
 - b. SELECT IdFilm, Titre
FROM Film
WHERE IdFilm NOT IN (SELECT IdFilm FROM Voter)

Partie B : Systèmes, technologies et Internet

Le formulaire HTML ci-dessous, permet de saisir les informations relatives à un réalisateur donné, puis, d'appeler la page "**Insertion.php**" pour les insérer dans la table appropriée.

| | |
|--|---|
| Id réalisateur : | <input type="text" value="122"/> |
| Nom réalisateur : | <input type="text" value="JamesCameron"/> |
| <div><input type="button" value="Annuler"/> <input type="button" value="Valider"/></div> | |

1. Expliquer le principe de communication entre le formulaire HTML et la page "**Insertion.php**".
2. Lors de l'envoi des données du formulaire vers la page "**Insertion.php**", le navigateur affiche, dans sa barre d'adresse, les informations suivantes :

http:// Insertion.php?IdRes=122&NomRes=JamesCameron

- a. Quelle est la méthode d'envoi utilisée dans ce formulaire ?
- b. Compléter le script de la page "**Insertion.php**" ci-dessous par les instructions nécessaires à la récupération et l'insertion des données.

```
< ?php

// On suppose que les connexions au serveur Web et à la base de données sont déjà établies

$id=.....[" ....."] ; // instruction de récupération de l'identifiant du réalisateur.

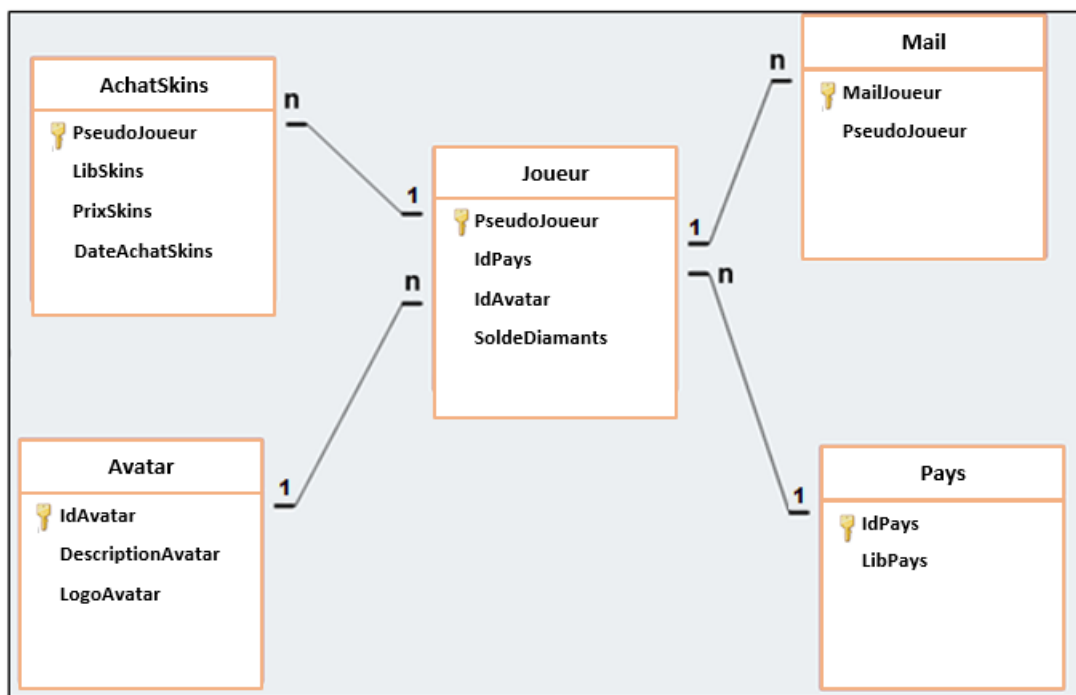
$nom=.....[" ..... "] ; // instruction de récupération du nom du réalisateur.

// Instruction(s) permettant l'insertion dans la base des données reçues
.....
.....
.....
.....
?>
```



Exercice N°4

Cet exercice porte sur la gestion de bases de données

Ci-dessous un extrait de la représentation graphique d'une base de données qui gère les utilisateurs d'une plateforme d'un jeu en ligne :



Lexique et description :

-  : représente une clé primaire.
-  : représente une relation entre tables.
- **Avatar** : un personnage de base utilisé dans le jeu.
- **Skin** : des modifications (habillement, effets visuels, lumière, arme, etc.) apportées à un avatar.

| Champ | Description |
|-------------------|---|
| PseudoJoueur | Pseudonyme d'un joueur |
| IdPays | Identifiant du pays d'origine d'un joueur. |
| IdAvatar | Identifiant d'un avatar |
| SoldeDiamants | Nombre de diamants en possession d'un joueur, utilisé pour l'achat des skins. |
| DescriptionAvatar | Description d'un avatar |
| LogoAvatar | Image d'un avatar |

| Champ | Description |
|----------------|---|
| MailJoueur | Email d'un joueur |
| LibPays | Libellé du pays d'un joueur |
| LibSkins | libellé d'un skin |
| PrixSkins | Prix d'un skin exprimé en nombre de diamants. N.B. : ce prix est fixe pour un skin. |
| DateAchatSkins | Date d'achat d'un skin |

Ci-dessous deux extraits relatifs aux tables **Joueur** et **AchatSkins** :

Table **Joueur**

| PseudoJoueur | IdMail | IdPays | IdAvatar | SoldeDiamants |
|--------------|--------|--------|----------|---------------|
| ak21so | 1 | Tu | 1 | 0 |
| sam210 | 2 | Tu | 1 | 200 |
| boss15 | 2 | Fr | 1 | 120 |
| abb2 | 21 | Tu | 2 | 10 |
| olfa | 3 | Tu | 1 | 14 |
| majed | 2 | Fr | 2 | 5 |

Table **AchatSkins**

| PseudoJoueur | LibSkins | PrixSkins | DateAchatSkins |
|--------------|---------------|-----------|----------------|
| olfa | masque zoro | 2 | 2020-07-01 |
| majed | costume ninja | 3 | 2020-10-03 |
| sam210 | costume ninja | 3 | 2021-01-05 |
| boss15 | masque zoro | 2 | 2021-01-06 |
| abb2 | costume ninja | 3 | 2021-03-16 |

On se propose de corriger la représentation de cette base de données en respectant les règles de gestion suivantes :

- **R1** : Un joueur ne peut pas acheter le même skin plusieurs fois.
- **R2** : Un joueur peut acheter plusieurs skins.
- **R3** : Plusieurs joueurs peuvent acheter le même skin.
- **R4** : Un joueur admet une seule adresse mail.
- **R5** : Plusieurs joueurs peuvent choisir le même avatar.

Questions :

1. Quelles sont les modifications à apporter à la représentation de la base de données afin de respecter la règle **R4**.
2. Soient les requêtes **A1** et **A2** suivantes :
A1 : Insert into AchatSkins values ('ak21so' , 'masque zoro' , 2 , 2021-08-04)
A2 : Insert into AchatSkins values ('sam210' , 'masque zoro' , 2 , 2021-10-12)

- a. Compléter le tableau ci-dessous en précisant pour chaque requête si l'insertion de l'enregistrement est réalisée ou non. Justifier votre réponse.

| Requête | Insertion réalisée (Oui/Non) | Justification |
|-----------|---------------------------------|---------------|
| A1 | | |
| A2 | | |

- b. En déduire la règle de gestion non respectée.
3. Les informations relatives aux Skins (LibSkins et PrixSkins) dans la table **AchatSkins** présentent une anomalie.
 - a. Identifier cette anomalie.
 - b. Proposer une solution pour remédier à cette anomalie.
4. À la suite des anomalies détectées et les rectifications réalisées, dresser la nouvelle représentation graphique ou textuelle de la base de données afin de respecter toutes les règles de gestion citées précédemment.
5. Afin de déterminer l'âge des joueurs, on se propose d'ajouter à la table **Joueur** (déjà créée) un champ intitulé "**DateNais**" de type Date. Écrire la requête SQL permettant d'effectuer cette tâche.
6. Pour s'assurer que le champ "**SoldeDiamants**" ne peut contenir que des valeurs positives, écrire la requête SQL permettant d'ajouter cette contrainte.