2022

Test technique

Développeur d’applications PHP / Bases de données / Système



**Nom du candidat :** ATTENTION : document corrigé

**Date du test :** ATTENTION : document corrigé

# Partie bases de données MySQL – cas 1

Nous considérerons pour l’ensemble de cette partie les tables suivantes :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table « **livre** » :   |  |  |  |  | | --- | --- | --- | --- | | livre\_id | livre\_prix | livre\_titre | livre\_genre | | 1 | 40 | Le glaive magique | 1 | | 3 | 40 | Gaffes en gros | 1 | | 4 | 40 | Lagaffe nous gâte | 1 | | 5 | 45 | QRN sur Bretzelburg | 1 | | 6 | 80 | Tour de manège | 3 | | 7 | 45 | Le spectre aux balles d'or | 1 | | 8 | 30 | La bonne chanson | 2 | | 9 | 50 | La jeune Parque | 2 | | 10 | 50 | Michel Strogoff | 4 | | 11 | 50 | La Serpe d'or | 1 | | 12 | 70 | Toujours aimer | 3 | | 13 | 70 | Toujours aimante | 3 | | 14 | 72 | Toujours aimé | 3 | | Table « **lib\_genre** » :   |  |  | | --- | --- | | genre\_id | genre\_libelle | | 1 | Bande Dessinée | | 2 | Poésie | | 3 | Roman de gare | | 4 | Roman | | 5 | Science-Fiction | |

## Effectuer la requête permettant d’afficher le livre numéro 7

Réponse :

Réponse : SELECT \* FROM livres WHERE livre\_id = 7;

## Effectuer la requête permettant d’afficher les livres supérieurs à 50€

Réponse :

Réponse : SELECT \* FROM livres WHERE libre\_prix > 50;

## Effectuer la requête permettant d’afficher les livres comprennant la chaine « tou »

Réponse :

Réponse : SELECT \* FROM livres WHERE livre\_titre LIKE 'tou%';

## Effectuer la requête permettant d’afficher les poésies supérieurs à 40€

Réponse :

Réponse : SELECT \* FROM livres WHERE livre\_genre = 2 AND livre\_prix > 40';

## Donner le **résultat** de cette requête :

## « SELECT livre\_titre, genre\_libelle FROM livre

## INNER JOIN lib\_genre ON livre\_genre = genre\_id

## WHERE livre\_prix IN (72,80) ; »

Réponse :

Réponse :

* Toujours aimé - Roman de gare
* Tour de manège - Roman de gare

## Effectuer la requête permettant d’afficher le nombre de livres par genre et le total de prix par cette catégorie

Réponse :

Réponse : SELECT COUNT(\*), SUM(livre\_prix) FROM livres GROUP BY livre\_genre;

# Partie bases de données MySQL – cas 2

Nous considérerons pour l’ensemble de cette partie les tables suivantes :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table « **eleves** »   |  |  |  | | --- | --- | --- | | id | nom | prenom | | 1 | Dupont | Albert | | 2 | Martin | Sophie | | 3 | Boucher | Alain | | Table « **notes** » :   |  |  |  | | --- | --- | --- | | id\_eleve | matiere | note | | 1 | MATHS | 12.4 | | 1 | HISTOIRE | 7.5 | | 1 | PHYSIQUE | *NULL* | | 2 | MATHS | 8 | | 2 | HISTOIRE | 11 | | 3 | MATHS | *NULL* | | 3 | HISTOIRE | 13.5 | |

## Lister pour chaque élève sa moyenne dans chacune des matières

Réponse :

Réponse : SELECT nom, prenom, AVG(note) FROM eleves INNER JOIN notes ON id=id\_eleve GROUP BY id\_eleve;

## Afficher les informations de l’élève ayant la plus grosse note en histoire

Réponse :

Réponse : SELECT nom, prenom, MAX(note) FROM eleves INNER JOIN notes ON id=id\_eleve WHERE matiere = ‘HISTOIRE’;

## Qu’est ce qu’une clé primaire ? Quelle est son utilitée ?

Réponse :

## A quoi servent les index ? Dans quel cas doit-on les utiliser ?

Réponse :

# Partie PHP procédural

## Remplacer le mot « World » par « Universe »  via une fonction PHP :

$string = "Hello World !" ;

Réponse :

Réponse : str\_replace("World", "Universe", $string);

## Afficher le mombre de caractère de cette chaine, puis l’afficher en majuscule.

$string = "Hello World !" ;

Réponse :

Réponse : echo strlen($string); echo strtoupper($string);

## Soit la déclation suivante :

$a = 15 ; $b = 12 ; $c = 1 ;

## Que va afficher :

if($a == 15) echo 'hello';

else echo 'world';

Réponse :

Réponse : hello

## Que va afficher :

echo ($b < 12) ? 'hello' : 'world';

Réponse :

Réponse : world

## Que va afficher :

if($c <= 10 && $b%2 != 0) echo 'hello';

else echo 'world';

Réponse :

Réponse : world

## Que va afficher :

if($a >= $b + $c)

echo 'hello' ;

else

echo 'world' ;

Réponse :

Réponse : hello

## Réaliser le code PHP qui permet d’afficher cette liste au format html sachant que les élèves sont stockés dans un tableau nommé $aEleve :

<ul>

<li class=’impair’>Eleve 1</li>

<li >Eleve 2</li>

<li class=’impair’>Eleve 3</li>

….

</ul>

Réponse :

# Partie PHP orienté objet

## Soit la déclation suivante :

class eleve{

var $id;

var $nom;

var $prenom;

public function \_\_construct() {

$this->nom = "dupont";

$this->prenom = "jean";

}

public function f1() {

$this->nom = "perret";

}

private function f2() {

$this->prenom = "marc";

}

protected function f3() {

$this->nom = "perret";

$this->prenom = "marc";

}

}

## Que va afficher :

$my\_eleve = new eleve();

echo $my\_eleve->nom.' '.$my\_eleve->prenom.'<br/>';

Réponse :

Réponse : dupont jean

## Que va afficher :

$my\_eleve = new eleve();

$my\_eleve->f1();

echo $my\_eleve ->nom.' '.$my\_eleve ->prenom.'<br/>';

Réponse :

Réponse : perret jean

## Que va afficher :

$my\_eleve = new eleve();

$my\_eleve->f2();

echo $my\_eleve ->nom." ".$my\_eleve ->prenom.'<br/>';

Réponse :

Réponse : Fatal error: Call to private method

## Que va afficher :

$my\_eleve = new eleve();

$my\_eleve->f3();

echo $my\_eleve ->nom ." ".$my\_eleve ->prenom.'<br/>';

Réponse :

Réponse : Fatal error: Call to protected method

# Connaissances générales

## Donner une définition au modèle « MVC ». Quels moyens techniques permettent de le réaliser ?

Réponse :

## Quels sont les bonnes pratiques que vous connaissez dans l’intégration d’email ?

Réponse :

## Réaliser un formulaire qui demande le nom, le prénom, la date de naissance, si la personne procède un animal (boutons radio), son pays d’origine (select) et qui transmettra le tout sur une page enregistrement.php

Réponse :

**Comment procédez-vous pour mettre à jour la structure d’une BDD par rapport à vos entités ?**

Il faut impérativement utiliser doctrine et ca commande associées :   
php app/console doctrine :schema :update –force ( --dump-sql pour afficher les requêtes sql executées)  
On peut utiliser create à la place de update la première fois.

**Quel est l’intérêt d’un service dans symfony2 ?**

De structurer et ranger le code afin d’en faciliter la maintenance.  
Egalement de rendre chaque fonction indépendante les unes des autres et pouvoir les faire évoluer indépendamment sans risquer de casser le fonctionnement d’un autre service (principe SOLID)