

Nom : .....

Prénom : .....

N° d'étudiant : .....

Examen du 16 décembre 2019 de  
**Technologies Web Avancées**

Master DDL — Parcours DILIPEM  
 Université Grenoble-Alpes

**Durée** 60 minutes.

**Supports** Tous documents autorisés.

**Remarque** Les questions munies d'un signe ♣ peuvent avoir zéro, une ou plusieurs réponses correctes. Les autres questions n'ont qu'une réponse correcte.

**Le barème est sur 23, mais des points négatifs peuvent être attribués pour les très mauvaises réponses.**

Les questions et réponses aux questions n'apparaissent pas dans le même ordre sur toutes les copies...

## 1 Git (4,5 pts)

**Question 1 ♣** (2.5 pts) Vous travaillez sur un programme informatique et vous venez de changer la structure de votre base de données `structure.sql` ainsi que le programme qui l'interroge `queries.php`. En outre, vous avez créé un meilleur suivi des erreurs dans "`errorHandling.php`".

Vous voulez **préparer** un commit uniquement pour ce qui touche à la base de données. Quelle(s) commande(s) devez-vous utiliser ?

- `git init`
- `git add structure.sql queries.php`
- `git push origin master`
- `git add structure.sql`  
`git add queries.php`  
`git commit -m`
- `git add .`  
`git reset errorhandling.php`
- `git add .`
- `git commit -m "update DB"`
- `git add structure.sql`  
`git add queries.php`  
`git commit -m "update DB"`
- `git add *.sql`
- `git add structure.sql`  
`git add queries.php`
- Aucune de ces réponses n'est correcte.

**Explication :** On veut uniquement préparer le commit et non le faire. Seules les commandes qui permettent de (dé)sélectionner les modifications sont attendues.

## CORRECTION

**Question 2** (2 pts) Vous venez de taper `git push origin master` et avez obtenu le message suivant.

```
loizbek@fartolva:~/Bureau/tasseaux$ git push origin master
Décompte des objets: 30, fait.
Delta compression using up to 4 threads.
Compression des objets: 100% (30/30), fait.
Écriture des objets: 100% (30/30), 6.84 KiB | 411.00 KiB/s, fait.
Total 30 (delta 11), reused 0 (delta 0)
To gricad-gitlab.univ-grenoble-alpes.fr:lzbk/sapinpinpin.git
 * [new branch]      master -> master
loizbek@fartolva:~/Bureau/tasseaux$ git add .
loizbek@fartolva:~/Bureau/tasseaux$ git commit -m "mise en forme du code"
[master fd364d1] mise en forme du code
 1 file changed, 2 insertions(+), 2 deletions(-)
loizbek@fartolva:~/Bureau/tasseaux$ git push origin master
To gricad-gitlab.univ-grenoble-alpes.fr:lzbk/sapinpinpin.git
 ! [rejected]        master -> master (fetch first)
error: impossible de pousser des références vers 'git@gricad-gitlab.univ-grenoble-alpes.fr:lzbk/sapinpinpin.git'
astuce: Les mises à jour ont été rejetées car la branche distante contient du travail que
astuce: vous n'avez pas en local. Ceci est généralement causé par un autre dépôt poussé
astuce: vers la même référence. Vous pourriez intégrer d'abord les changements distants
astuce: (par exemple 'git pull ...') avant de pousser à nouveau.
astuce: Voir la 'Note à propos des avances rapides' dans 'git push --help' pour plus d'information.
```

Que s'est-il passé ?

- Vous avez oublié de faire votre `git commit` et vous n'avez rien à ajouter au dépôt distant.
- Vous avez fait votre `git commit` sans préalablement sélectionner de modifications avec `git add`.
- Vous avez oublié de donner un intitulé à votre commit.
- Entre la dernière fois où vous avez fait un `git pull` et votre `git push`, quelqu'un a fait un `git push` sur le serveur

## 2 Questions de cours (4 pts)

**Question 3** (1 pt) La commande pour modifier les droits d'accès d'un fichier ou un dossier est :

- `chgrp`
- `git commit`
- `cp`
- `chmod`
- `chown`

**Question 4** ♣ (3 pts) Moodle est un LMS (*Learning Management System*), c'est à dire un CMS (*Content Management System*) spécialisé dans l'apprentissage. Sans représenter l'unique stratégie de développement possible, il met en place un ensemble de stratégies de développement que l'on pourra retrouver dans de nombreux systèmes de ce type.

Sélectionnez les éléments qui vous paraissent corrects :

- Tous les fichiers du dossier moodle peuvent être supprimés et retéléchargés depuis le site de moodle sans difficulté.
- L'historique des actions effectuées sur la plateforme est stocké dans la base de données.
- Moodle nécessite une base de données pour fonctionner.
- Le code php est exécuté par le navigateur de la personne qui se connecte au site.
- Les mots de passe sont stockés en clair dans la base de données. Cela permet de faciliter la récupération des comptes si un utilisateur perd ses informations.
- Les fichiers envoyés sont stockés dans un dossier spécifié par l'administrateur.
- Aucune de ces réponses n'est correcte.

**Explication :**

- Le fichier de configuration contient des informations qui sont propres à chaque installation.
- La table '`*_logstore_standard_log`' contient toutes les actions effectuées.

### 3 Problème : critique de Pâtisserie

Un critique pâtissier a décidé de se créer une base de données pour recenser les spécialités qu'il goûte.

Il ajoute un [Artisan](#) à la base de données lorsqu'il planifie sa première visite. En revanche il ne sait pas à l'avance ce qu'il va goûter et n'intègre les pâtisseries que lorsqu'il a fait sa dégustation et écrit sa critique.

Il tient à décrire chaque pâtisserie indépendamment pour pouvoir ensuite savoir quel artisan fait la meilleure qu'il ait goûtée. Par exemple, il veut être sûr de pouvoir trouver facilement le meilleur « pain au chocolat » qu'il ait goûté. Pour s'assurer, qu'il n'a pas appelé « Pain au chocolat » chez l'un et « chocolatine » chez l'autre la même pâtisserie, l'interface de saisie lui propose d'ajouter des pâtisserie uniquement s'il n'a pas trouvé la pâtisserie dans une liste déroulante.

Le schéma de la base de données est fourni en annexe p. 6.

#### 3.1 Vocabulaire (3 pts)

**Question 5** (1 pt) Dans la base de données modélisée dans le schéma de la partie 4, « '`Patisserie`'. '`idPat`' » est :

- un index
- une clé unique
- un enregistrement
- une clé primaire
- une clé étrangère
- une table

**Question 6** (1 pt) Dans la base de données modélisée dans le schéma de la partie 4, « 'Specialite'. 'idPat' » est :

- une clé étrangère
- une clé unique
- un index
- une table
- une clé primaire
- un enregistrement

**Question 7** (1 pt) Dans la base de données modélisée dans le schéma de la partie 4, « 'Ville' » est :

- un attribut
- une colonne
- une requête
- une table
- une clé primaire
- un enregistrement

### 3.2 Cardinalités (4 pts)

On rappelle que les cardinalités de relation se lisent de la manière suivante :

- « Un artisan a une boutique dans une ville et une seule »
- « Une ville contient 1 ou plusieurs artisans »

**Question 8** (1 pt) Dans la base de données modélisée dans le schéma de la partie 4, la cardinalité "A" (de la relation [Artisan-Pâtisserie](#)) est :

- 1..n
- 1
- 0..n

**Explication :** Chaque pâtisserie est ajoutée en relation avec un Artisan. En revanche, plusieurs pâtisseries peuvent proposer un même gâteau (ex : mille-feuille).

**Question 9** (1 pt) Dans la base de données modélisée dans le schéma de la partie 4, la cardinalité "B" (de la relation [Artisan-Pâtisserie](#)) est :

- 1
- 0..n
- 1..n

**Explication :** L'artisan est ajouté avant la dégustation, il est donc possible qu'aucune spécialité ne lui soit associée.

## CORRECTION

**Question 10** (2 pts) D'après les cardinalités, l'une des tables n'est pas nécessaire et pourrait être supprimée. Laquelle ?

- Patisserie : contient des informations qui pourraient toutes être stockées dans la table Specialite.
- Artisan-Ville : contient des informations qui pourraient toutes être intégrées à la table Artisan
- Specialite : contient des informations que l'on pourrait intégrer à la table Patisserie
- Artisan : contient des informations qui pourraient toutes être intégrées à la table Patisserie
- Ville : contient des informations qui pourraient toutes être stockées dans la table Artisan

### 3.3 Requêtes (6,5 ±0,5 pts)

**Question 11** (1,5 pts) Le critique vient de goûter un pain au chocolat (IdPat=23) à 95 cts, chez l'artisan dont le numéro de téléphone est le 05 55 55 55 55 et veut lui mettre la note de 5 :

- `INSERT INTO 'Specialite' SET ('IdPat', 'TelArtisan', 'Prix', 'Note')VALUES (23, '05 55 55 55 55', '0.95', 5)`
- `UPDATE 'Specialite' SET 'Prix'='0.95', 'Note'=5 WHERE 'IdPat'='23'AND 'TelArtisan'='05 55 55 55 55',`
- Sans le texte de la critique, on ne peut rien faire.
- `INSERT INTO 'Specialite' ('IdPat', 'TelArtisan', 'Prix', 'Note')VALUES (23, '05 55 55 55 55', '0.95', 5)`
- `INSERT INTO 'Specialite' ('IdPat', 'TelArtisan', 'Note', 'Prix')VALUES (23, '05 55 55 55 55', '0.95', 5)`

**Question 12** (1,5 pts) Le critique voudrait connaître l'ensemble des pâtisseries qu'il a goûtées :

- `INSERT INTO 'Patisseries' ('Nom', 'Description', 'Type')VALUES ('idPat', 'DISTINCT')`
- `SELECT 'Nom', 'Description', 'Type' FROM 'Patisseries'`
- `SELECT * FROM 'Specialite'`
- `SELECT 'Nom', 'Description', 'Type' FROM 'Patisserie'`
- `SELECT * FROM 'Specialite' WHERE 'idPat' != 0`

**Question 13** (1,5 pts) Un artisan change de numéro de téléphone et du 06 66 66 66 66, il passe au 07 77 77 77 77 :

- Sans le nom de l'artisan, on ne peut rien faire.
- `UPDATE 'Artisan' SET 'Telephone' = '06 66 66 66 66'WHERE 'Telephone' = '07 77 77 77 77'`
- `INSERT INTO 'Artisan' ('Telephone')VALUES ('07 77 77 77 77')`
- `UPDATE 'Artisan' SET 'Telephone' = '07 77 77 77 77'WHERE 'Telephone' = '06 66 66 66 66'`

## CORRECTION

**Question 14** (2 pts) Le critique veut la liste des artisans chez lesquels il a goûté des pains au chocolat (idéalement il aimerait les afficher triés par ordre de ville croissant) ainsi que la note qu'ils ont obtenue. Écrivez une requête qui lui permette de le faire.

---

---

---

---

---

RÉPONDRE SUR LES LIGNES NE PAS UTILISER LES CASES CI-DESSOUS

2,5     2     1,5     1     0,5     0

**Explication :**

```
1 SELECT 'Artisan'.'Nom' as 'Pâtissier',
2        'Artisan'.'NomBoutique' as 'Boutique',
3        'Ville'.'Nom' as 'Lieu',
4        'Specialite'.'Note' as 'Note'
5 FROM 'Artisan'
6      JOIN 'Artisan-Ville' ON 'Artisan-Ville'.'TelArtisan'='Artisan'.'Telephone'
7      JOIN 'Ville' ON 'Artisan-Ville'.'IdVille'='Ville'.'IdVille'
8      JOIN 'Specialite' ON 'Specialite'.'TelArtisan'='Artisan'.'Telephone'
9 WHERE 'Specialite'.'IdPat'=23
10 ORDER BY 'Lieu' ASC, 'Note' DESC ;
```

## 4 Annexe : Schéma de la base de données pour la partie 3

