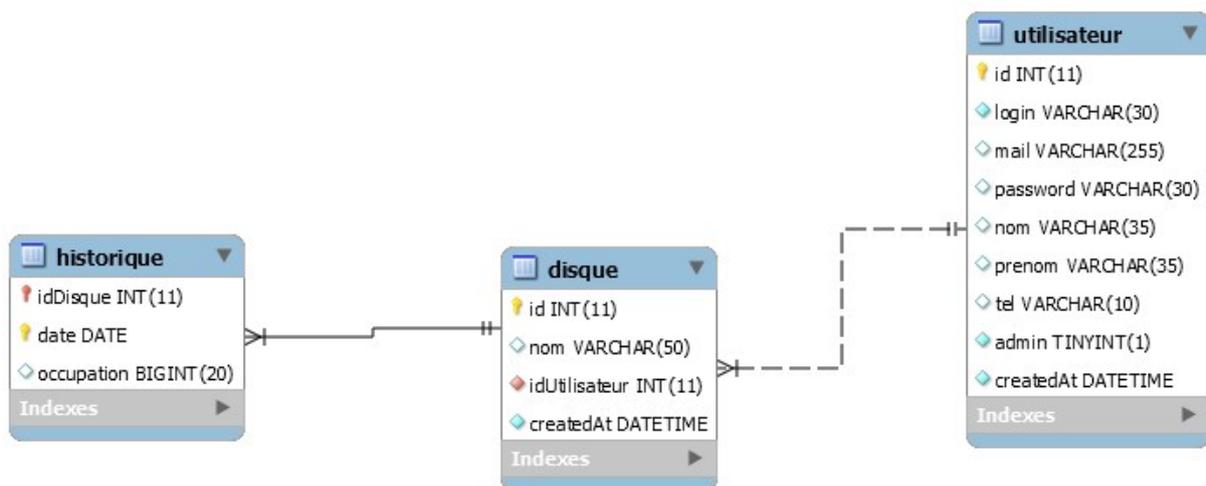


Préparation à E5	2 <sup>ème</sup> BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°1
Durée max	1 heure 30
Domaines abordés	D4.1 - Conception et réalisation d'une solution applicative
Fichier	DS1.docx

## Partie 1

### UML :

A partir de l'extrait du schéma physique de la base de données **Cloud** :



1. Représentez le diagramme de classes UML correspondant.
2. Implantez les classes en ne spécifiant que les membres de données.

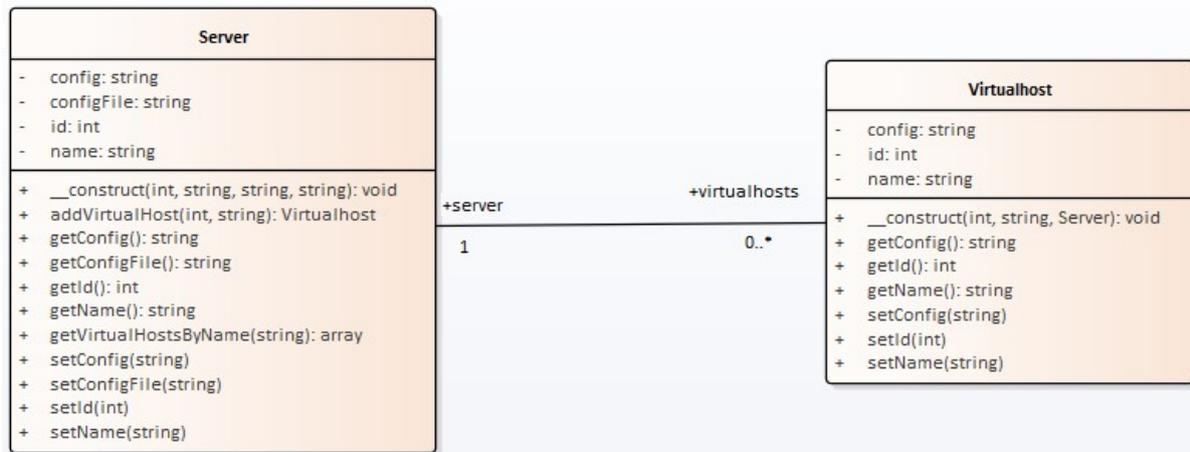
## Partie 2

### Implémentation à partir du projet virtualhosts

A partir du diagramme de classes en page suivante, implémentez les méthodes ci-dessous de la classe **Server** :

3. Le modificateur **setName** et l'accessor **getName** de la propriété **name**.
4. Le Constructeur **\_\_construct**
5. La méthode **addVirtualhost**
6. La méthode **getVirtualHostsByName**

Préparation à E5	2 <sup>ème</sup> BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°1



```

class Server{

    private $id;
    private $name;
    private $config;
    private $configFile;
    /**
     * @var Collection de Virtualhost
     */
    private $virtualhosts;

    /**
     * Constructeur de la classe, initialise tous les membres de données
     * @param int $id
     * @param string $name
     * @param string $config
     * @param string $configFile
     */
    public function __construct($id,$name,$config,$configFile){}

    /**
     * Crée un nouveau virtual host et l'ajoute aux virtualhosts du serveur
     * @param int $id
     * @param string $name
     * @return Virtualhost
     */
    public function addVirtualhost($id,$name){}

    /**
     * Retourne la liste des virtualhosts du serveur dont le nom est égal à
    celui passé en paramètre
     * @param string $name nom recherché
     * @return Collection de Virtualhost
     */
    public function getVirtualHostsByName($name){}
}

class Virtualhost{

    private $id;
    private $name;
    private $config;
  
```

Préparation à E5	2 <sup>ème</sup> BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°1

```

/**
 * @var Server
 */
private $server;

/**
 * Constructeur de la classe, initialise tous les membres
 * @param int $id
 * @param string $name
 * @param Server $server
 */
public function __construct($id,$name,$server){...}

public function getName(){...}
public function getId(){...}
public function getConfig(){...}
public function getServer(){...}
}

/**
 * Classe technique Collection
 */
class Collection{
/**
 * Retourne le nombre d'objets dans la collection
 * @return int
 */
public function cardinal (){...}
/**
 * Vide la collection de ses objets
 */
public function vider(){...}
/**
 * Retourne l'objet à l'index $index (à partir de 0)
 * @param int $index
 * @return Object
 */
public function obtenirObjet($index){...}
/**
 * Ajoute l'objet $objet dans la collection
 * @param Object $objet
 */
public function ajouter($objet){...}
/**
 * Supprime l'objet $objet de la collection
 * @param Object $objet
 */
public function supprimer($objet){...}
}

```

Pour instancier une collection :        \$uneCollection = new Collection() de <classe>

Pour parcourir par itération tous les éléments d'un objet Collection :

```

foreach( $variable_objet in $uneCollection){
    // instructions avec $variable_objet
}

```