



<b>Centre organisateur :</b> <b>UNIVERSITÉ D'AVIGNON</b>	<b>Épreuve d'admissibilité</b>
<b>Session 2018</b>	<b><u>Durée</u> : 3 heures</b> <b><u>Coefficient</u> : 4</b>
<b><u>Concours</u> : Assistant Ingénieur</b> <b><u>BAP</u> : E (Informatique, Statistique et Calcul Scientifique)</b> <b><u>Nature</u> : externe</b> <b><u>Spécialité</u> : Gestionnaire application / Assistance support</b>	<b>30/05/2018</b> <b>13h30 – 16h30</b>

### **Consignes :**

- L'usage de la calculatrice n'est pas autorisé. L'usage du téléphone portable est interdit. Tout document et autres matériels électroniques sont interdits.
- Vérifier, dès la remise du sujet, que celui-ci comporte bien **10** pages, y compris celle-ci et les annexes.
- La qualité de la rédaction ainsi que l'orthographe seront pris en compte dans la notation.

### **Attention !**

Il vous est rappelé que :

- Votre identité ne doit figurer que dans la partie du coin en-tête de la copie (ou des copies) mise(s) à votre disposition.
- Toute mention d'identité portée sur toute autre partie de la copie (ou des copies) que vous remettrez en fin d'épreuve provoquera l'annulation de votre épreuve.

## **I – Questions générales (3 points)**

1. Qu'est-ce qu'une authentification forte ?
2. Que signifie le RGPD ?
3. Qu'est-ce qu'une GED ?
4. À quoi sert un référentiel de données ?
5. À quoi sert un Reverse Proxy ?
6. À quoi sert un helpdesk ?

## **II – Connaissances métier (6 points)**

1. Qu'est-ce qu'un serveur d'application? Décrire ses principales fonctions? Citez deux noms de serveurs d'application parmi les plus connus
2. Quelles sont les propriétés ACID ? Quels sont leurs buts ?
3. À quoi servent les logs binaires dans une base de données ?
4. Qu'est-ce que le polymorphisme ?
5. Donnez les avantages et les inconvénients d'un moteur de template ?
6. Sur un système Linux, donnez la commande permettant de modifier le propriétaire du répertoire « application » et son contenu par l'utilisateur « www-data » ?

### III – Compétences opérationnelles (5 points)

Soit la table PERSONNEL créée dans une base de données relationnelle

```
CREATE TABLE PERSONNEL (  
  NOM          VARCHAR(20),  
  PRENOM      VARCHAR(20),  
  NAISSANCE   DATE,  
  MAIL       VARCHAR(80)  
);
```

(N.B : syntaxe MySQL en annexe)

1. Écrire les instructions qui permettent d'insérer dans la table PERSONNEL les personnes suivantes :
  - Dupond Albert né le 22/03/1959 dont l'email est [albert.dupond@free.fr](mailto:albert.dupond@free.fr)
  - Dupond Marie née le 10 décembre 1970 dont l'email est [marie.dupond@yahoo.fr](mailto:marie.dupond@yahoo.fr)
  - Martin Sophie née le 2 janvier 1987 dont l'email est [msophie@gmail.com](mailto:msophie@gmail.com)
2. Écrire l'instruction qui permet de modifier le prénom de Mme Marie Dupond en Mme Maria Dupond.
3. Écrire en PL/SQL une fonction qui à partir du nom de la personne renvoie son email  
N.B : en cas de doublons ou homonyme, cette fonction devra traiter une exception (exemple cas de Dupond).
4. Écrire les instructions SQL permettant :
  - de modifier la table PERSONNEL pour y rajouter une colonne supplémentaire (ID\_AGENT)
  - de définir cette colonne (ID\_AGENT) comme clé primaire de la table PERSONNEL.
5. À quoi sert un trigger ? Quelles sont les similitudes et/ou différences par rapport à une procédure stockée ? Écrire un trigger qui à chaque insertion de tuple dans cette table incrémente la valeur de la colonne ID\_AGENT.

## IV - Étude de cas (6 points)

On vous demande de réfléchir à la mise en place d'une application web qui permet d'effectuer une démarche de candidature au sein de votre établissement. Elle permettra de créer un compte utilisateur afin d'y déposer son CV et sa lettre de motivation.

Potentiellement, cette application sera sollicitée de façon importante et nécessitera une haute disponibilité.

L'interface se compose de deux vues :

- La première correspond au formulaire d'inscription (un prénom, un nom, un identifiant, un mot de passe, une date de naissance et un mail)
- La seconde vue est une zone de dépôt de fichiers qui nécessite une authentification. Cette vue permet de déposer des fichiers au format PDF (CV et lettre de motivation). L'utilisateur doit choisir également le service auquel il souhaite candidater. Les services ne sont pas en champ libre.

Pour améliorer la sécurité des informations, l'identifiant doit être unique et permet d'authentifier un utilisateur. Un message d'erreur sera soulevé si l'identifiant existe déjà en base de données.

1. Définir le modèle entité-association répondant à cet énoncé. Puis en déduire son modèle relationnel (structure des tables et leurs relations).
2. Écrire un algorithme permettant l'insertion d'un utilisateur dans la base de données depuis le formulaire d'inscription. Cet algorithme doit permettre de garantir l'unicité d'un utilisateur.
3. Proposer grâce à un schéma une architecture web permettant de répondre aux contraintes de haute disponibilité (on ne demande pas ici de répondre en détail, mais d'avoir une idée globale)
4. Votre correspondant métier pour ce projet est le service des ressources humaines. Décrire la manière dont vous procéderiez pour garantir l'adhésion de cette nouvelle application.

## Annexe 1 : extrait de la documentation MySQL 5.7

### 14.1.18 CREATE TABLE Syntax

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [table_options]
    [partition_options]
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)]
    [table_options]
    [partition_options]
    [IGNORE | REPLACE]
    [AS] query_expression
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    { LIKE old_tbl_name | (LIKE old_tbl_name) }
create_definition:
    col_name column_definition
    | [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...)
      [index_option] ...
    | {INDEX|KEY} [index_name] [index_type] (index_col_name,...)
      [index_option] ...
    | [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY]
      [index_name] [index_type] (index_col_name,...)
      [index_option] ...
    | {FULLTEXT|SPATIAL} [INDEX|KEY] [index_name] (index_col_name,...)
      [index_option] ...
    | [CONSTRAINT [symbol]] FOREIGN KEY
      [index_name] (index_col_name,...) reference_definition
    | CHECK (expr)
column_definition:
    data_type [NOT NULL | NULL] [DEFAULT default_value]
    [AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
    [COMMENT 'string']
```

```

[COLUMN_FORMAT {FIXED|DYNAMIC|DEFAULT}]
[STORAGE {DISK|MEMORY|DEFAULT}]
[reference_definition]
| data_type [GENERATED ALWAYS] AS (expression)
  [VIRTUAL | STORED] [UNIQUE [KEY]] [COMMENT comment]
  [NOT NULL | NULL] [[PRIMARY] KEY]

```

data\_type:

```

  BIT[(length)]
| TINYINT[(length)] [UNSIGNED] [ZEROFILL]
| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
| INT[(length)] [UNSIGNED] [ZEROFILL]
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| BIGINT[(length)] [UNSIGNED] [ZEROFILL]
| REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
| FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| DATE
| TIME[(fsp)]
| TIMESTAMP[(fsp)]
| DATETIME[(fsp)]
| YEAR
| CHAR[(length)] [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| VARCHAR(length) [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| BINARY[(length)]
| VARBINARY(length)
| TINYBLOB
| BLOB

```

```

| MEDIUMBLOB
| LONGBLOB
| TINYTEXT [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]
| TEXT [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]
| MEDIUMTEXT [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]
| LONGTEXT [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]
| ENUM(value1,value2,value3,...)
    [CHARACTER SET charset_name] [COLLATE collation_name]
| SET(value1,value2,value3,...)
    [CHARACTER SET charset_name] [COLLATE collation_name]
| JSON
| spatial_type
index_col_name:
    col_name [(length)] [ASC | DESC]
index_type:
    USING {BTREE | HASH}
index_option:
    KEY_BLOCK_SIZE [=] value
| index_type
| WITH PARSER parser_name
| COMMENT 'string'
reference_definition:
    REFERENCES tbl_name (index_col_name,...)
    [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]
    [ON DELETE reference_option]
    [ON UPDATE reference_option]
reference_option:
    RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT

```

table\_options:

table\_option [[,] table\_option] ...

table\_option:

ENGINE [=] engine\_name

| AUTO\_INCREMENT [=] value

| AVG\_ROW\_LENGTH [=] value

| [DEFAULT] CHARACTER SET [=] charset\_name

| CHECKSUM [=] {0 | 1}

| [DEFAULT] COLLATE [=] collation\_name

| COMMENT [=] 'string'

| COMPRESSION [=] {'ZLIB'|'LZ4'|'NONE'}

| CONNECTION [=] 'connect\_string'

| DATA DIRECTORY [=] 'absolute path to directory'

| DELAY\_KEY\_WRITE [=] {0 | 1}

| ENCRYPTION [=] {'Y' | 'N'}

| INDEX DIRECTORY [=] 'absolute path to directory'

| INSERT\_METHOD [=] { NO | FIRST | LAST }

| KEY\_BLOCK\_SIZE [=] value

| MAX\_ROWS [=] value

| MIN\_ROWS [=] value

| PACK\_KEYS [=] {0 | 1 | DEFAULT}

| PASSWORD [=] 'string'

| ROW\_FORMAT [=] {DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}

| STATS\_AUTO\_RECALC [=] {DEFAULT|0|1}

| STATS\_PERSISTENT [=] {DEFAULT|0|1}

| STATS\_SAMPLE\_PAGES [=] value

| TABLESPACE tablespace\_name [STORAGE {DISK|MEMORY|DEFAULT}]

| UNION [=] (tbl\_name[,tbl\_name]...)

partition\_options:

PARTITION BY

{ [LINEAR] HASH(expr)

| [LINEAR] KEY [ALGORITHM={1|2}] (column\_list)

```

| RANGE{(expr) | COLUMNS(column_list)}
| LIST{(expr) | COLUMNS(column_list)}
[PARTITIONS num]
[SUBPARTITION BY
  { [LINEAR] HASH(expr)
  | [LINEAR] KEY [ALGORITHM={1|2}] (column_list) }
[SUBPARTITIONS num]
]
[(partition_definition [, partition_definition] ...)]

```

partition\_definition:

```

PARTITION partition_name
[VALUES
  {LESS THAN {(expr | value_list) | MAXVALUE}
  |
  IN (value_list)}]
[[STORAGE] ENGINE [=] engine_name]
[COMMENT [=] 'comment_text' ]
[DATA DIRECTORY [=] 'data_dir']
[INDEX DIRECTORY [=] 'index_dir']
[MAX_ROWS [=] max_number_of_rows]
[MIN_ROWS [=] min_number_of_rows]
[TABLESPACE [=] tablespace_name]
[(subpartition_definition [, subpartition_definition] ...)]

```

subpartition\_definition:

```

SUBPARTITION logical_name
[[STORAGE] ENGINE [=] engine_name]
[COMMENT [=] 'comment_text' ]
[DATA DIRECTORY [=] 'data_dir']
[INDEX DIRECTORY [=] 'index_dir']
[MAX_ROWS [=] max_number_of_rows]
[MIN_ROWS [=] min_number_of_rows]
[TABLESPACE [=] tablespace_name]

```

query\_expression:

```
SELECT ... (Some valid select or union statement)
```

#### 14.1.20 CREATE TRIGGER Syntax

CREATE

```
[DEFINER = { user | CURRENT_USER }]
```

```
TRIGGER trigger_name
```

```
trigger_time trigger_event
```

```
ON tbl_name FOR EACH ROW
```

```
[trigger_order]
```

```
trigger_body
```

trigger\_time: { BEFORE | AFTER }

trigger\_event: { INSERT | UPDATE | DELETE }

trigger\_order: { FOLLOWS | PRECEDES } other\_trigger\_name

#### 14.1.21 CREATE VIEW Syntax

CREATE

```
[OR REPLACE]
```

```
[ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
```

```
[DEFINER = { user | CURRENT_USER }]
```

```
[SQL SECURITY { DEFINER | INVOKER }]
```

```
VIEW view_name [(column_list)]
```

```
AS select_statement
```

```
[WITH [CASCADED | LOCAL] CHECK OPTION]
```

#### 14.7.3.1 CREATE FUNCTION Syntax for User-Defined Functions

```
CREATE [AGGREGATE] FUNCTION function_name RETURNS {STRING|INTEGER|REAL|DECIMAL}
```

```
SONAME shared_library_name
```