

Nom : RABOBA

Prénom : Tokinantenaina

Antonio Sandrica

Programme D-CLIC

Spécialité : Backend

Évaluation Intermédiaire –
MySQL, base de donnée d' une
librairie.

Date : 01/07/2022

I– Analyse et correction de
la base de données.

Les tâches :

1. Chargement de la base
2. Les corrections :
 - a. Modification du code

a. 1– On a ajouté DEFAULT CHARACTER SET utf8 :

```
CREATE DATABASE biblio DEFAULT  
CHARACTER SET utf8 ;
```

Pour lire les caractères avec
‘accents’ .

a. 2– Les valeurs par défaut

La valeur de la date d’ emprunt
‘dateEmp’ de ‘biblio.emprunter’
peut être égale à la valeur de la
date au moment de
l’ enregistrement : ‘NOW ()’ .

On attribue la valeur par défaut
d’ une durée maximal d’ emprunt
par exemple ‘14’ jours.

Ces deux valeurs par défaut (date
d’ emprunt et durée maximal)
permettent de contrôler le livre
pour qu’ il ne soit pas trop
longtemps emprunté entraînant une
perte de la livre, en essayant de
programmer une notification quand
la durée est passée.

```
ALTER TABLE `biblio`.`emprunter`  
CHANGE COLUMN `dateEmp` `dateEmp` DATE  
NOT NULL DEFAULT NOW() ,  
CHANGE COLUMN `dureeMax` `dureeMax` INT(11)  
NOT NULL DEFAULT 14 ;
```

b. TRIGGER

Ajout d' un TRIGGER pour mettre le
‘Nom’ en Capital. (Cette action
et d' autre vérification pourront
être faites depuis la saisie au
niveau du Front)

3. Les tuples pour chaque table

Formule :

```
SELECT COUNT('clé primaire') FROM 'nom de table'
```

4. Nombres des attributs pour chaque table

```
SELECT COUNT(*) FROM  
INFORMATION_SCHEMA.COLUMNS WHERE  
TABLE_NAME = 'nom de table '
```

5. Affichage des clés primaire pour chaque table

```
SELECT TABLE_NAME as 'nom de la table',  
GROUP_CONCAT(COLUMN_NAME) as 'Clé primaire'  
FROM  
INFORMATION_SCHEMA.KEY_COLUMN_USAGE  
WHERE  
TABLE_SCHEMA = 'biblio'  
AND CONSTRAINT_NAME='PRIMARY'  
GROUP BY TABLE_NAME;
```

II- Interactions avec la base de données

1. Les livres empruntés actuellement

```
SELECT DISTINCT livres.NL as 'Livre  
actuellement emprunté', livres.NO,  
oeuvres.titre, oeuvres.auteur  
FROM emprunter, adherents, livres, oeuvres  
WHERE dateRet IS NULL  
and adherents.NA = emprunter.NA  
and livres.NL = emprunter.NL and oeuvres.NO =  
livres.NO
```

2. Les livres empruntées par 'Jeanette Lecoœur'

```
SELECT adherents.NA, nom, prenom, NL,  
dateEmp FROM adherents inner join emprunter  
where adherents.NA = emprunter.NA and  
prenom='Jeanette' and nom = 'Lecoœur'
```

3. Les livres empruntées en septembre 2009

```
SELECT NL, dateEmp FROM biblio.emprunter
WHERE dateEmp BETWEEN '2009-09-1' AND
'2009-09-31' ORDER by dateEmp
```

4. Tous les adhérents qui ont emprunté un livre de 'Fedor Dostoievski'

```
SELECT distinct emprunter.NA, adherents.nom,
pre nom, tel, emprunter.NL, livres.NO, titre,
auteur
FROM adherents, emprunter, livres, oeuvres
where adherents.NA = emprunter.NA and
emprunter.NL = livres.NL and
livres.NO = oeuvres.NO
and
auteur LIKE "%Fedor Dostoievski%"
```

5. 13/. 14/. 15/. Base de données mise à jour

6. 16/

M. Cyril FREDERIC essaye
d' emprunter le livre n° 23.

Requête :

```
INSERT INTO `biblio`.`emprunter` (`NL`, `NA`)
VALUES ('23', '28');
```

```
ERROR 1062: 1062: Duplicate entry  
'23-0000-00-00' for key 'PRIMARY'
```

Cette requête produit une erreur car le livre n° 23 n' est plus encore disponible. Il vient d' être emprunter par Martine CROZIER. On constate que les livres empruntés vont tous renvoyer une erreur de requête même s' ils ont rendues (avec date de retour *not null*)
Donc il faut penser à créer un TRIGGER qui interagira avec la table 'emprunter' et une autre table 'Historique' en supprimant de la table 'emprunter' les livres avec un 'date de retour' *non null* et de le mettre dans la nouvelle table 'Historique' .

7. 17/ Succès
8. 18/ le ou les auteurs du titre « Voyage au bout de la nuit »

```
SELECT auteur, titre FROM biblio.oeuvres  
where titre = "Voyage au bout de la nuit"
```

9. 19/

Quels sont les ou les éditeurs du
titre « Narcisse et Goldmund »

```
SELECT distinct livres.NO, titre, auteur as 'de',  
editeur as 'Editeur'  
FROM oeuvres, livres where oeuvres.NO =  
livres.NO  
and titre = « Narcisse et Goldmund »
```

10. 20/

Quels sont les adhérents
actuellement en retard ?

```
SELECT adherents.*, (DATE_ADD(dateEmp,  
INTERVAL dureeMax DAY)) as 'DateRetPrevu',  
DATEDIFF((DATE_ADD(dateEmp, INTERVAL  
dureeMax DAY)), curdate()) as 'duree passée (j)'  
FROM biblio.emprunter, adherents where  
adherents.NA = emprunter.NA and dateRet is  
null  
and DATEDIFF((DATE_ADD(dateEmp,  
INTERVAL dureeMax DAY)), curdate()) < 0
```

11. 21/

Quels sont les livres actuellement
en retard ?


```

SELECT distinct livres.NL, oeuvres.NO,
dateEmp, titre, auteur, (DATE_ADD(dateEmp,
INTERVAL dureeMax DAY)) as 'DateRetPrevu',
DATEDIFF((DATE_ADD(dateEmp, INTERVAL
dureeMax DAY)), curdate()) as 'duree passée (j)'
FROM emprunter, livres, oeuvres
where oeuvres.NO = livres.NO and
emprunter.NL = livres.NL and dateRet is null
and DATEDIFF((DATE_ADD(dateEmp,
INTERVAL dureeMax DAY)), curdate()) < 0

```

12. 22/

Les adhérents en retard

```

SELECT distinct adherents.NA, nom, prenom,
dateEmp, dureeMax, dateRet,
(DATE_ADD(dateEmp, INTERVAL dureeMax
DAY)) as 'DateRetPrevu',
(CASE
    WHEN (
(DATEDIFF((DATE_ADD(dateEmp, INTERVAL
dureeMax DAY)), dateRet)) is null)
    THEN
DATEDIFF((DATE_ADD(dateEmp, INTERVAL
dureeMax DAY)), curdate())
    WHEN
((DATEDIFF((DATE_ADD(dateEmp, INTERVAL
dureeMax DAY)), dateRet)) is not null)
    THEN
DATEDIFF((DATE_ADD(dateEmp, INTERVAL
dureeMax DAY)), dateRet)
END) as 'duree passée (j)'
FROM emprunter, adherents where
adherents.NA = emprunter.NA
and DATEDIFF((DATE_ADD(dateEmp,
INTERVAL dureeMax DAY)), dateRet) < 0

```

or adherents.NA = emprunter.NA and dateRet is null
and DATEDIFF((DATE_ADD(dateEmp,
INTERVAL dureeMax DAY)), curdate()) < 0

13. 23/ Nombre de livres empruntées par auteur.

```
SELECT distinct auteur, count(livres.NO) as  
'nombre de livre emprunter par auteur'  
FROM livres, oeuvres  
where oeuvres.NO = livres.NO  
GROUP BY oeuvres.NO order by  
count(livres.NO);
```

14. 24/ Nombre de livres empruntés par éditeur.

```
SELECT distinct auteur, count(livres.NO) as  
'nombre de livre emprunter par auteur'  
FROM livres, oeuvres  
where oeuvres.NO = livres.NO  
GROUP BY oeuvres.NO order by  
count(livres.NO);
```

15. 25/ Durée moyenne des emprunts rendus.

```
SELECT (sum(datediff(dateEmp, dateRet)) /  
count(emprunter.NL))  
as "Durée moyenne de retour des livres rendus  
(j)"
```

```
FROM biblio.emprunter  
where dateRet is not null
```

16. 26/ Durée moyenne des retards pour l'ensemble des emprunts.

*creation d' un view :

```
CREATE  
ALGORITHM = UNDEFINED  
DEFINER = `root` @`localhost`  
SQL SECURITY DEFINER  
VIEW durée des livres en retard AS  
SELECT DISTINCT  
    adherents.NA AS NA,  
    adherents.nom AS nom,  
    adherents.prenom AS prenom,  
    emprunter.dateEmp AS dateEmp,  
    emprunter.dureeMax AS dureeMax,  
    emprunter.dateRet AS dateRet,  
    emprunter.dateEmp + INTERVAL  
    emprunter.dureeMax DAY AS DateRetPrevu,  
    CASE  
        WHEN TO_DAYS(emprunter.dateEmp +  
            INTERVAL emprunter.dureeMax DAY) -  
            TO_DAYS(emprunter.dateRet) IS NULL THEN  
            TO_DAYS(emprunter.dateEmp + INTERVAL  
                emprunter.dureeMax DAY) -  
            TO_DAYS(CURDATE())  
        WHEN TO_DAYS(emprunter.dateEmp +  
            INTERVAL emprunter.dureeMax DAY) -  
            TO_DAYS(emprunter.dateRet) IS NOT NULL  
        THEN TO_DAYS(emprunter.dateEmp +
```

```

INTERVAL emprunter.dureeMax DAY) -
TO_DAYS(emprunter.dateRet)
    END AS jPassed
FROM
    (emprunter
    JOIN adherents)
WHERE
    adherents.NA = emprunter.NA
    AND TO_DAYS(emprunter.dateEmp +
INTERVAL emprunter.dureeMax DAY) -
TO_DAYS(emprunter.dateRet) < 0
    OR adherents.NA = emprunter.NA
    AND emprunter.dateRet IS NULL
    AND TO_DAYS(emprunter.dateEmp +
INTERVAL emprunter.dureeMax DAY) -
TO_DAYS(CURDATE()) < 0

```

Puis,

```

SELECT abs((sum(jPassed)/count(NA))) as
'durée moyenne des retards emprunts' FROM
biblio.`durée des livres en retard`;

```

17. 27/ Durée moyenne des retards parmi les seuls retardataires.

```

SELECT abs((sum(`duree passée
(j)`)/count(NL)))
as 'durée moyenne des retards emprunts actuel'
FROM
biblio.`durée_livres_retardés_actuellement`;

```