



Université de Nouakchott Al Aasriya  
Faculté des Sciences et Techniques  
Département Mathématiques et Informatiques



---

## Mini-Projet de TP :



# Création de Cron Jobs pour la Sauvegarde Automatique de Bases de Données

---

Réalisé par :

Mohamed Lemine Abdallahi Tah

C12896

---

Encadré par :

Dr Cheikhne Mohamed Mahmoud

Année universitaire : 2024 – 2025

*Projet réalisé dans le cadre du Master SSD – Statistiques et Sciences des Données*

# Table des matières

Introduction générale	2
1 Outils utilisés	2
2 Création de la base de données de test	2
3 Script de sauvegarde de la base de données	3
4 Mise en place de la tâche planifiée (Cron)	3
5 Vérification de sauvegarde automatique	4
Conclusion	6

## Note\*

Ce projet porte sur le thème : « *Création de Cron Jobs pour la Sauvegarde Automatique de Bases de Données* ».

Les objectifs sont les suivants :

- Automatiser la sauvegarde quotidienne d'une base de données MySQL.
- Éviter la perte de données en cas de panne ou d'erreur humaine.
- Utiliser des scripts Bash et la planification via `cron` sous Ubuntu.

## Note\*

Le projet complet est disponible dans le dépôt GitHub suivant :

[https://github.com/MoLemineAb/cron\\_jobs](https://github.com/MoLemineAb/cron_jobs)

Le fichier `rapport_cron_jobs.pdf` contient la documentation détaillée du projet, tandis que le fichier `backup_univ.sh` contient le script Bash responsable de la sauvegarde automatique. Le répertoire contient les captures d'écran illustrant les tests . le fichier

`rapport.pdf` contient Rapport complet du projet en format PDF.

# Introduction

La sauvegarde régulière des bases de données est une pratique essentielle dans la gestion des systèmes d'information. Elle permet de garantir :

- la disponibilité des données en cas de panne,
- la protection contre les erreurs humaines ou logicielles,
- la continuité du service en cas de cyberattaque ou de perte de données.

Dans ce projet, nous avons automatisé la sauvegarde quotidienne d'une base MySQL sous Ubuntu en utilisant un script Bash exécuté via **cron**.

## 1 Outils utilisés

- **MySQL (XAMPP)** : Système de gestion de base de données relationnelle.
- **Ubuntu** : Système d'exploitation Linux utilisé pour l'environnement de développement.
- **Terminal Ubuntu** : Pour les commandes Bash, la crontab et les scripts.



FIGURE 1 – \*  
Logos des principaux outils utilisés

## 2 Création de la base de données de test

Dans le but de tester le système de sauvegarde automatique, nous avons créé une base de données fictive appelée **univ\_scolarite**. Elle représente un système simplifié de gestion des inscriptions universitaires.

Cette base de données est composée de trois tables principales :

- **etudiants** : contient les informations personnelles des étudiants comme le nom, le prénom, l'adresse email et la date de naissance.
- **cours** : contient les informations sur les cours dispensés, tels que le nom du cours, son code et l'enseignant responsable.
- **inscriptions** : relie les étudiants aux cours auxquels ils sont inscrits, avec la date d'inscription.

Ces tables sont liées logiquement entre elles à travers des identifiants, permettant de représenter les relations entre les étudiants, les cours et les inscriptions.

Cette base a été conçue dans le seul but de simuler un contexte réaliste pour les tests de sauvegarde. Quelques enregistrements ont été ajoutés manuellement pour valider le fonctionnement du script de sauvegarde.

### 3 Script de sauvegarde de la base de données

Afin d'automatiser la sauvegarde de la base de données **univ\_scolarite**, nous avons créé un script Bash nommé **backup\_univ.sh**.

Le script réalise notamment les actions suivantes :

- Création automatique d'un dossier destiné à stocker les fichiers de sauvegarde s'il n'existe pas déjà.
- Exportation de la base de données **univ\_scolarite** dans un fichier au format **.sql**.
- Ajout automatique de la date et de l'heure dans le nom du fichier pour distinguer chaque sauvegarde.
- Mise à jour d'un fichier **log** contenant un historique des sauvegardes effectuées (réussies ou échouées).

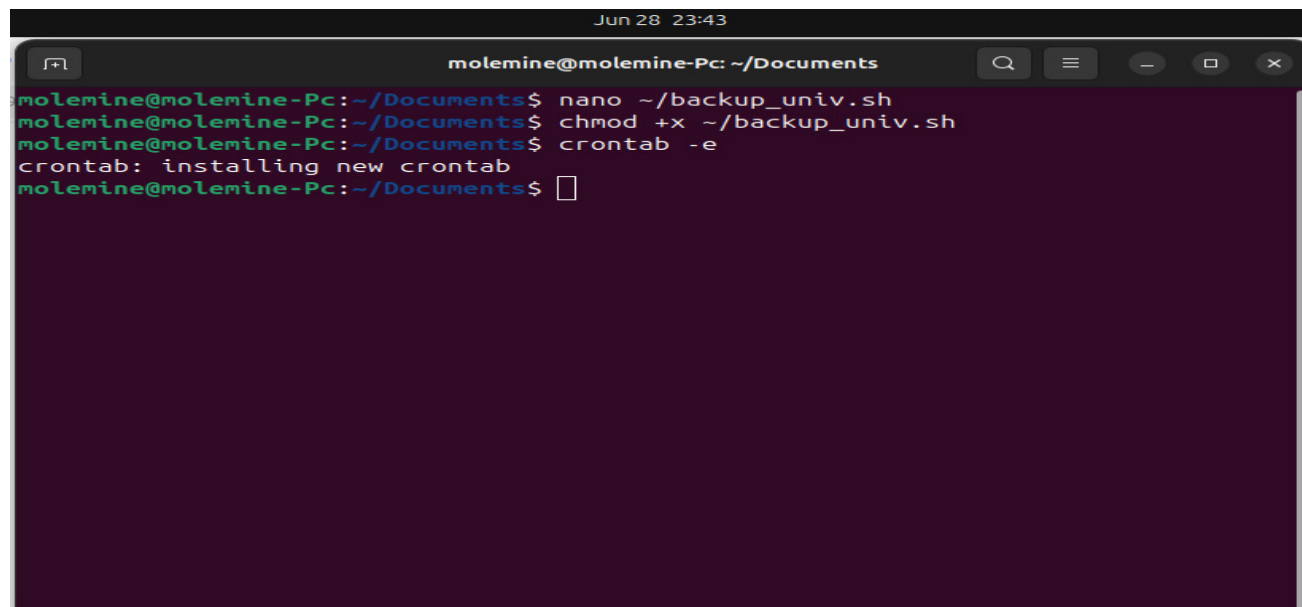
### 4 Mise en place de la tâche planifiée (Cron)

Après avoir rédigé le script de sauvegarde **backup\_univ.sh**, il est nécessaire de :

- lui attribuer les permissions d'exécution avec la commande **chmod +x**,
- puis d'ouvrir la table des tâches planifiées via la commande **crontab -e**.

Cela permet de configurer une exécution automatique du script tous les jours à une heure précise (dans notre cas, à **23h45**).

Voici une capture d'écran montrant les étapes réalisées dans le terminal Ubuntu :



```
Jun 28 23:43
molemine@molemine-Pc: ~/Documents
molemine@molemine-Pc:~/Documents$ nano ~/backup_univ.sh
molemine@molemine-Pc:~/Documents$ chmod +x ~/backup_univ.sh
molemine@molemine-Pc:~/Documents$ crontab -e
crontab: installing new crontab
molemine@molemine-Pc:~/Documents$
```

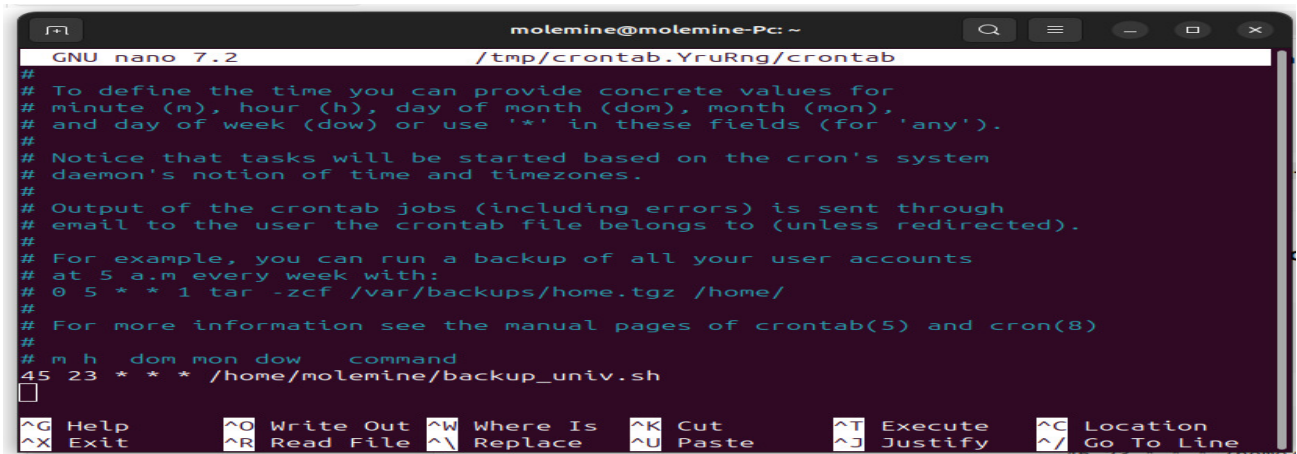
FIGURE 2 – Attribution des permissions et ouverture de la crontab

## Ajout de la tâche cron quotidienne

Après avoir ouvert la crontab avec la commande `crontab -e`, nous avons ajouté une ligne de configuration permettant d'exécuter le script de sauvegarde tous les jours à **23h45**.

Nous avons donc inséré la ligne suivante dans le fichier crontab :

```
45 23 * * * /home/molemine/backup_univ.sh
```

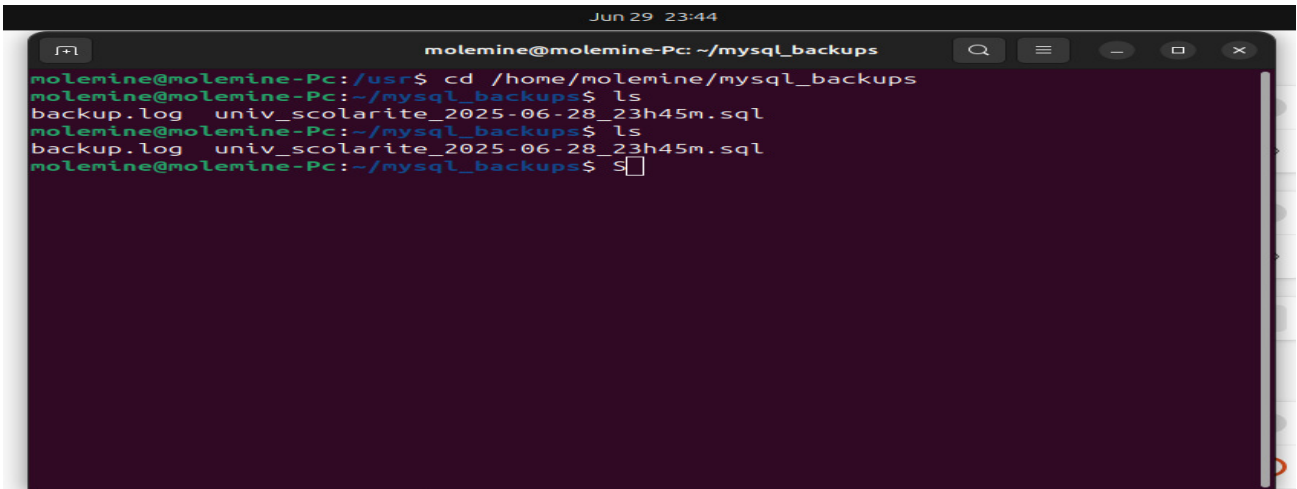


```
molemine@molemine-Pc: ~  
GNU nano 7.2 /tmp/crontab.YruRng/crontab  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
45 23 * * * /home/molemine/backup_univ.sh  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

FIGURE 3 – Ajout de la ligne dans la crontab pour la planification quotidienne

## 5 Vérification de sauvegarde automatique

Pour tester l'exécution automatique correcte du script, j'ai réalisé deux tests à dates différentes.



```
Jun 29 23:44  
molemine@molemine-Pc: ~/mysql_backups  
molemine@molemine-Pc:/usr$ cd /home/molemine/mysql_backups  
molemine@molemine-Pc:~/mysql_backups$ ls  
backup.log univ_scolarite_2025-06-28_23h45m.sql  
molemine@molemine-Pc:~/mysql_backups$ ls  
backup.log univ_scolarite_2025-06-28_23h45m.sql  
molemine@molemine-Pc:~/mysql_backups$ s
```

FIGURE 4 – Backup effectué automatiquement le 28/06/2025 à 23h45

### Premier test le 28 juin 2025

Dans un premier test le cron job s'est exécuté automatiquement, et un fichier de sauvegarde nommé : ( univ\_scolarite\_2025-06-28\_23h45m.sql ) a été généré dans le répertoire `mysql_backups`

## Deuxième test en avançant la date au 29 juin 2025

Pour réaliser un second test sans attendre 24 heures, j'ai volontairement modifié l'horloge du système en avançant la date à **29 juin 2025, 23h43**

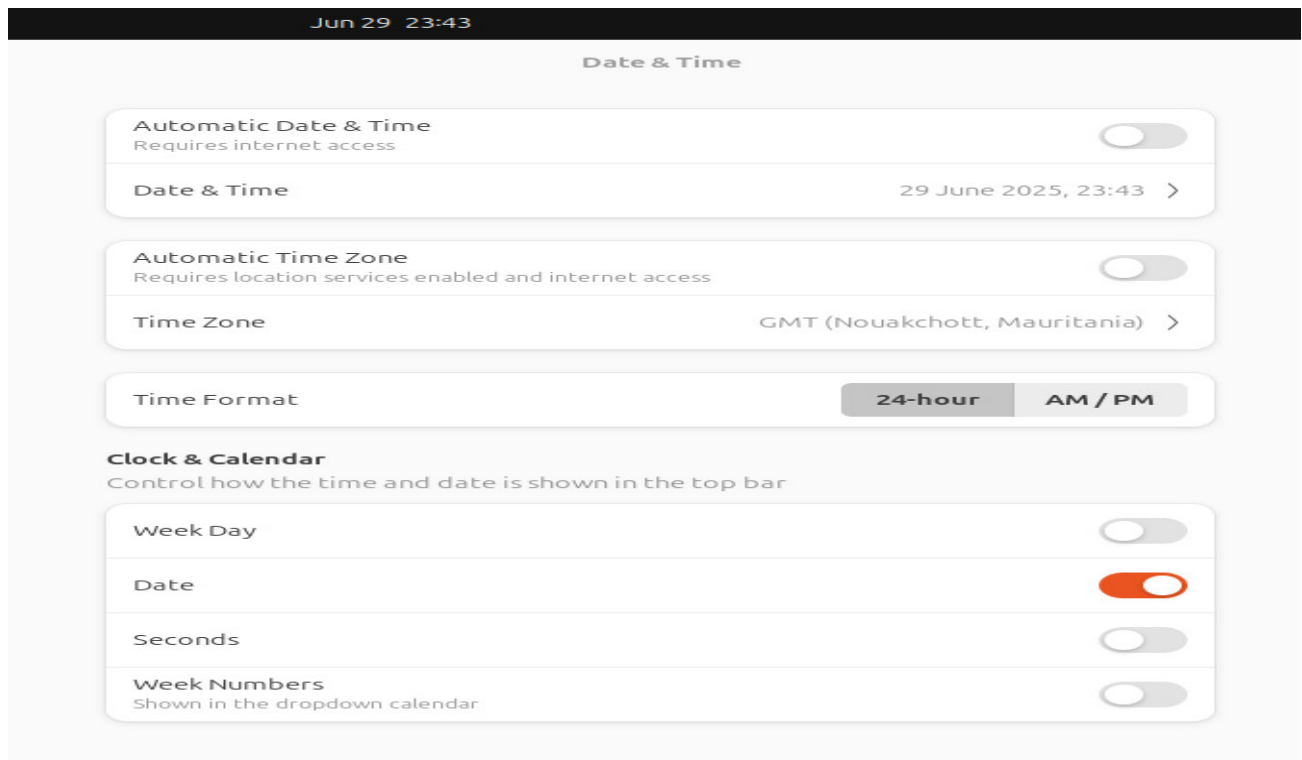


FIGURE 5 – Modification manuelle de la date système

Quelques minutes après, à **23h45**, le script s'est exécuté une nouvelle fois automatiquement. Le fichier suivant a été créé : ( `univ_scolarite_2025-06-29_23h45m.sql` ), ce qui prouve que le cron job fonctionne bien à chaque exécution quotidienne.

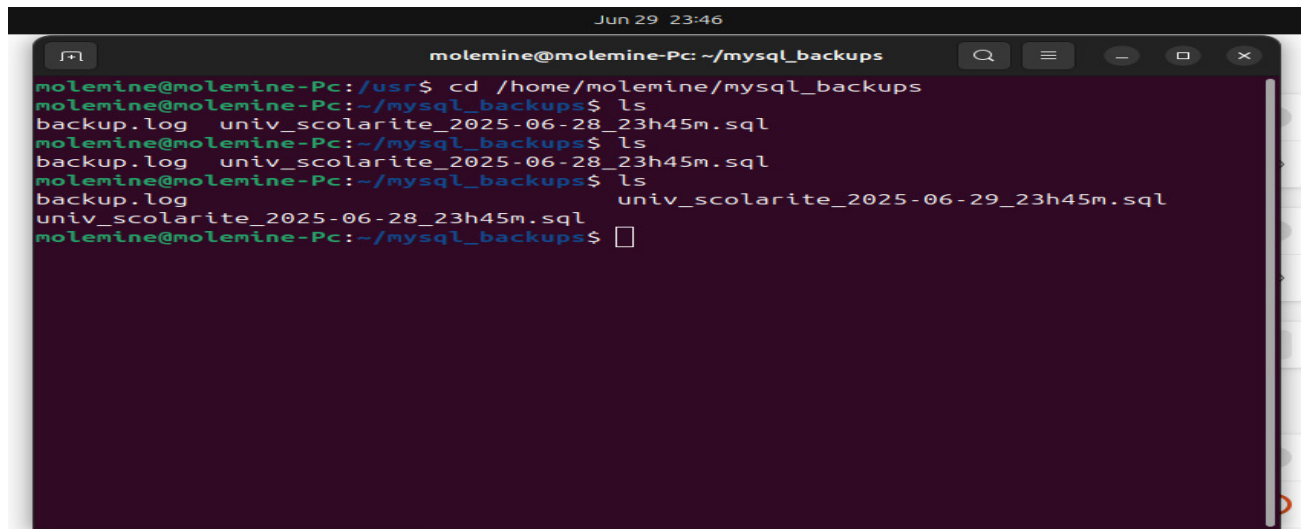


FIGURE 6 – Backup généré automatiquement le 29/06/2025 à 23h45

## Conclusion

Ce mini-projet nous a permis de mettre en pratique les notions de planification de tâches sous Linux en combinant l'utilisation de scripts Bash avec l'outil `cron`. Grâce à ce mécanisme, la sauvegarde automatique de la base de données `univ_scolarite` est désormais assurée quotidiennement, sans intervention manuelle.

L'approche utilisée ici peut facilement être adaptée à d'autres bases de données ou à des besoins de sauvegarde différents. En entreprise, ce type de solution contribue à renforcer la sécurité et la fiabilité des systèmes d'information.

Enfin, ce travail illustre l'importance de l'automatisation dans la gestion des infrastructures informatiques modernes.