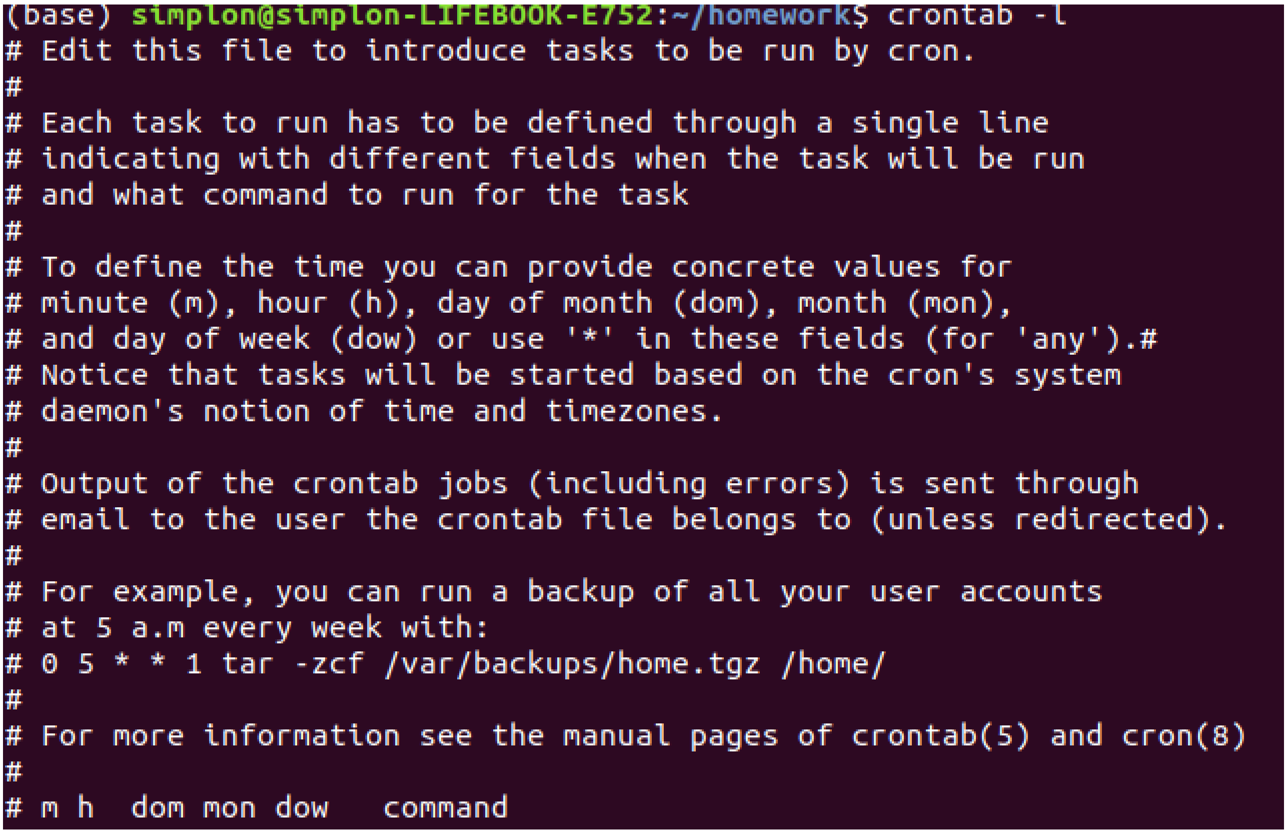
Contenu du script:

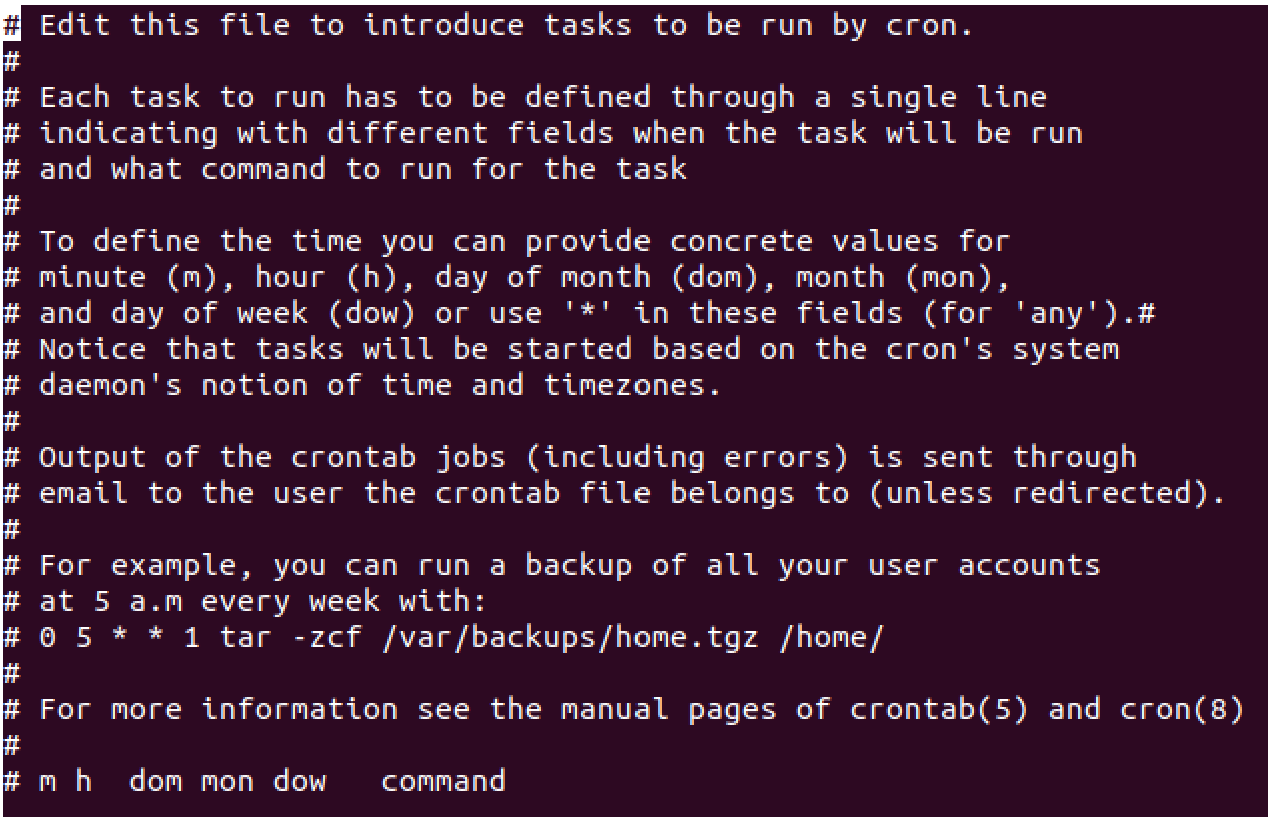
```
#!/bin/sh

# 1.Sauvegarder Sakila en utilisant mysqldump

# Save sakila without CREATE DATABASE (to allow to restore into another db)
echo "SAVE SAKILA"
mysqldump --user=AUTO_SAVE --password=Simplon_2020 --add-drop-table --routines --triggers sakila > /home/simplon/homework/save_sakila.sql


# 2.Sauvegarder toutes les bases de données en utilisant mysqldump

# Save all db
echo "SAVE ALL DB"
mysqldump --user=AUTO_SAVE --password=Simplon_2020 --add-drop-table --all-databases >  /home/simplon/homework/save_all.sql


# 3.Faire un check de la base de données netflix et Simplon en utilisant mysqlcheck

# Check simplon and netflix
echo "CHECK SIMPLON NETFLIX"
mysqlcheck --user=AUTO_SAVE --password=Simplon_2020 --databases simplon netflix > /home/simplon/homework/check_simplon_netflix_$(date +%Y-%m-%d-%H-%M-%S)


# 4.Faire un check de toutes les bases de données en utilisant mysqlcheck

# Check all db
echo "CHECK QLL DB"
mysqlcheck --user=AUTO_SAVE --password=Simplon_2020 --all-databases  > /home/simplon/homework/check_all_$(date +%Y-%m-%d-%H-%M-%S)


# 5.Restaurer le dump de Sakila sous un nom différent

# Create sakila_new if not exists
echo "create database if not exists sakila_new" | mysql -u AUTO_SAVE --password=Simplon_2020

# Restore sakila into sakila_new
echo "RESTORE SAKILA"
mysql -u AUTO_SAVE --password=Simplon_2020 -A -Dsakila_new < save_sakila.sql
```

crontab -l



```
(base) simplon@simplon-LIFEBOOK-E752:~/homework$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# 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
```

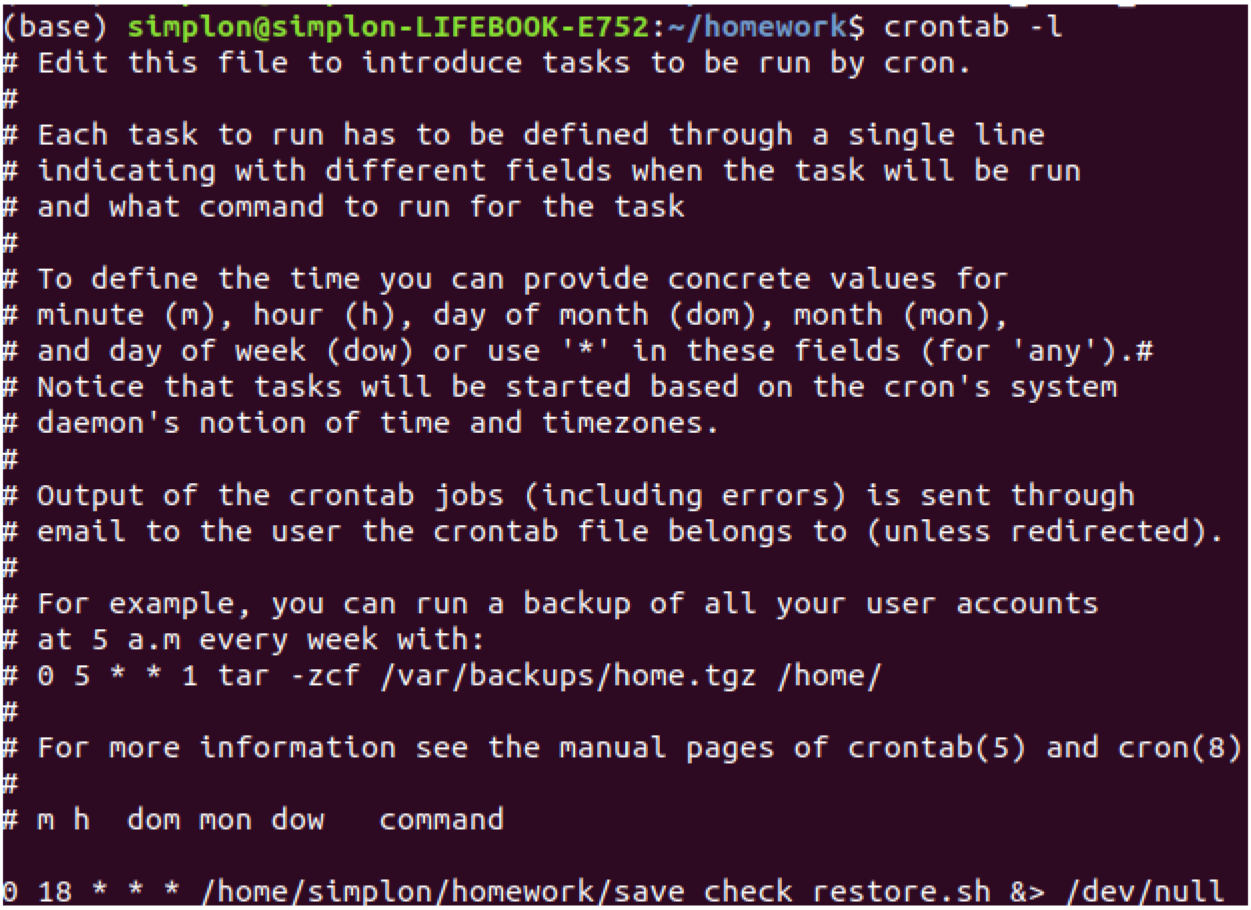
crontab -e



```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# 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
```

```
On rajoute selon la syntaxe indiquée:
0 18 * * * /home/simplon/homework/save_check_restore.sh &> /dev/null
(=> exécute le script tous les jours à 18h)
```

```
crontab -l  devient:
```



```
(base) simplon@simplon-LIFEBOOK-E752:~/homework$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# 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

0 18 * * * /home/simplon/homework/save check restore.sh &> /dev/null
```