Create the Backup Script in MySql/Job’s

First, create a script that will perform the backup and delete old backups. Save this script in a directory, for example /usr/local/bin/.

Backup_delete.sh

```
#!/bin/bash

# MySQL credentials
USER="your_username"
PASSWORD="your_password"
DATABASE="your_database"
BACKUP_DIR="/path/to/backup/directory"
RETENTION_DAYS=7

# Date format for the backup file
DATE=$(date +%Y%m%d%H%M)

# Backup file name
BACKUP_FILE="$BACKUP_DIR/$DATABASE-$DATE.sql"

# Create backup directory if it doesn't exist
mkdir -p $BACKUP_DIR

# Perform the backup
mysqldump -u $USER -p$PASSWORD $DATABASE > $BACKUP_FILE

# Check if the backup was successful
if [ $? -eq 0 ]; then
    echo "Backup successful: $BACKUP_FILE"
else
    echo "Backup failed"
    exit 1
fi

# Delete old backups
find $BACKUP_DIR -type f -name "$DATABASE-*.sql" -mtime +$RETENTION_DAYS -exec rm {} \;

# Check if the deletion was successful
if [ $? -eq 0 ]; then
    echo "Old backups deleted successfully"
else
    echo "Failed to delete old backups"
    exit 1
fi
```

Make the Script Executable

```
sudo chmod +x /usr/local/bin/backup_delete.sh
```

Step 2: Schedule the Script Using Cron

1. Open the crontab file for editing:
   ```
   crontab -e
   ```
2. Add a line to schedule the backup job. For example, to run the script every day at 2 AM:

```
0 2 * * * /usr/local/bin/backup_delete.sh >> /var/log/mysql_backup.log 2>&1
```

This cron job will run the backup_delete.sh script every day at 2 AM and log the output to /var/log/mysql_backup.log.

Explanation of Cron Syntax

- `0`: Minute when the job will run (0th minute).
- `2`: Hour when the job will run (2 AM).
- `*`: Day of the month (any day).
- `*`: Month (any month).
- `*`: Day of the week (any day).

Backp Delete Job Schedule Daily Basis

Step 1: Create the Stored Procedure

```
DELIMITER $$

CREATE PROCEDURE backup_daily_data()
BEGIN
    DECLARE backup_date DATE;
    SET backup_date = CURDATE() - INTERVAL 1 DAY;

    INSERT INTO backup_table (column1, column2, column3, backup_date)
    SELECT column1, column2, column3, backup_date
    FROM original_table
    WHERE DATE(record_timestamp) = backup_date;
END$$

DELIMITER ;
```

Step 2: Schedule the Stored Procedure Using MySQL Event Scheduler

```
CREATE EVENT daily_backup_event
ON SCHEDULE EVERY 1 DAY
STARTS (CURRENT_DATE + INTERVAL 1 DAY) AT '02:00:00'
DO
CALL backup_daily_data();
```

Step 3: Enable the MySQL Event Scheduler

Ensure that the MySQL Event Scheduler is enabled. You can enable it by adding the following line to your MySQL configuration file (my.cnf) and restarting MySQL, or by setting it dynamically.

Enable Event Scheduler in Configuration

```
[mysqld]
event_scheduler = ON
```

Enable Event Scheduler Dynamically

```
SET GLOBAL event_scheduler = ON;
```