***Scheduler (dagster)--->***

from dagster import schedule

from sp\_etl.jobs.run\_etl import run\_etl\_job

@schedule(cron\_schedule="35 22 17 \* \*", job=run\_etl\_job, execution\_timezone="US/Central")

def etl\_job\_schedule(\_context):

    run\_config = {}

    return run\_config

The function etl\_job\_schedule is a function that returns the run configuration for the run\_etl\_job when it’s executed according to the cron schedule specified in the @schedule decorator.

The @schedule decorator is used to define a schedule for running a Dagster job. In this case, it specifies that the run\_etl\_job should be executed according to the cron schedule "35 22 17 \* \*", which means that it will run at minute 35 past hour 22 (10:35 PM) on day-of-month 17 every month. The decorator also specifies that the job should be executed in the "US/Central" timezone.

When it’s time for the scheduled job to run, Dagster will call the etl\_job\_schedule function to get the run configuration for the run\_etl\_job. The run configuration is a dictionary of key-value pairs that specify how the job should be executed. In this case, an empty dictionary is being returned as the run configuration.

The purpose of defining a function like etl\_job\_schedule is to allow you to dynamically generate and return a run configuration for your scheduled job at runtime. This can be useful if you need to pass different arguments or configuration values to your job depending on when it’s being executed.

In this case, the first three fields represent the minute (35), hour (22), and day of the month (17) when the command should run. The two asterisks in the fourth and fifth fields mean that the command should run on any month and any day of the week.

1. from dagster import schedule: This imports the schedule decorator from the dagster module, which is used to define a scheduled job.
2. from sp\_etl.jobs.run\_etl import run\_etl\_job: This imports the run\_etl\_job function from the sp\_etl.jobs.run\_etl module. This function is the main function that will be executed when the scheduled job runs.
3. @schedule(cron\_schedule="35 22 17 \* \*", job=run\_etl\_job, execution\_timezone="US/Central"): This decorates the etl\_job\_schedule function with the schedule decorator. The cron\_schedule parameter specifies when the job should run, using a cron expression. In this case, the cron expression is 35 22 17 \* \*, which means the job will run at 10:35 PM on the 17th day of every month. The job parameter specifies the function that will be executed when the job runs, which is run\_etl\_job in this case. The execution\_timezone parameter specifies the timezone in which the job should be executed, which is "US/Central" in this case.
4. def etl\_job\_schedule(\_context):: This defines the etl\_job\_schedule function, which will be executed when the scheduled job runs. The \_context parameter is a required parameter for all Dagster functions, but it is not used in this case.
5. run\_config = {}: This initializes an empty run\_config dictionary, which will be used to pass configuration data to the run\_etl\_job function.
6. return run\_config: This returns the run\_config dictionary, which will be passed as a parameter to the run\_etl\_job function when it is executed.