Automating Tasks with the Scheduler

By Ahmed Baraka

Objectives

In this lecture, you will learn how to perform the following:

- Describe scheduler job attributes
- Create and manage time-based jobs
- Manage scheduler programs
- Manage scheduler schedules
- Monitor Scheduler jobs by querying dictionary views

Routine Job Examples

Run end of the month processing procedure

Calculate figures to be used by BI

Once a file is received in a directory, load it

Run a daily job to take backup of the database

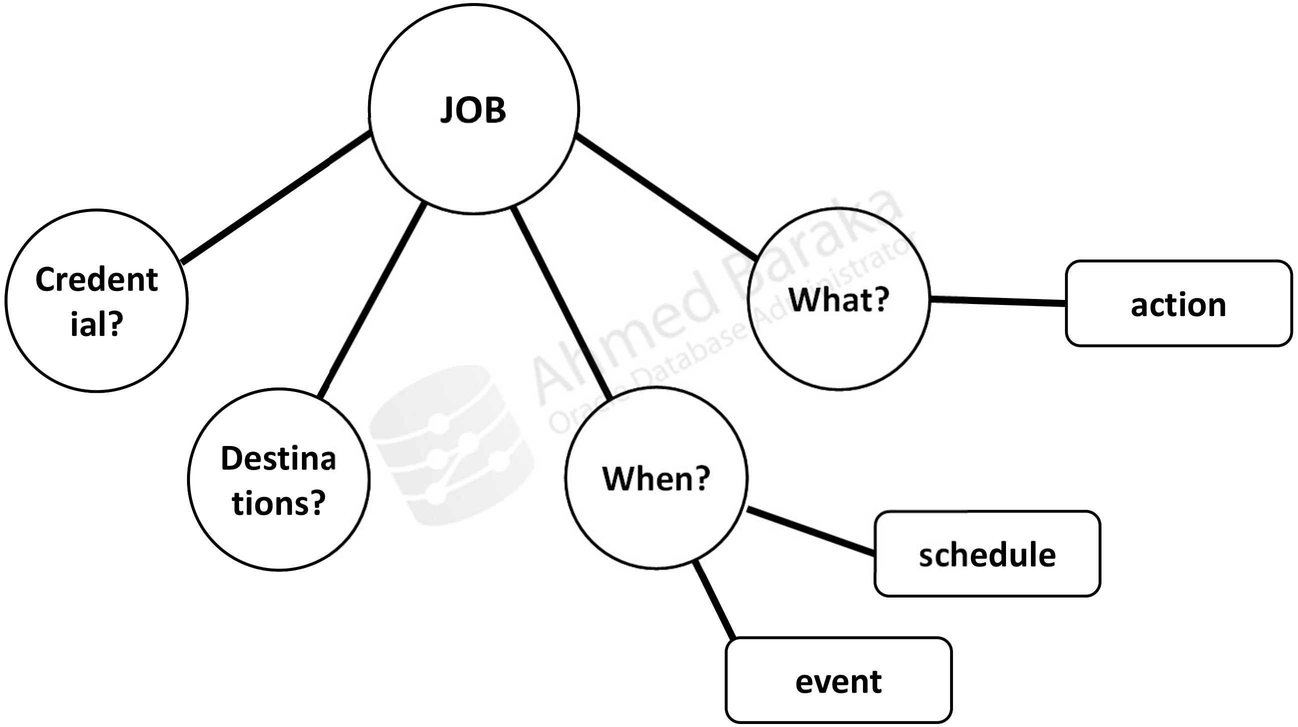
Rebuild application indexes every quarter

Generate complex reports on midnights

About the Scheduler

- Is used for scheduling and managing database jobs, tasks, and events in an Oracle database.
- Features:
  - Job scheduling
  - Job chaining
  - Resource management
  - Event-based scheduling
  - Job monitoring and reporting
  - Multiple job types
  - Remote execution

Scheduler Job



Creating a Scheduler Job

- To define a Scheduler job:
  - **Action**: what the job will run
    - Specify the job action attribute (**inline**): PL/SQL, Java, call OS scripts, run program
    - Setting an existing **Program**
  - **Schedule**: when the job starts
    - Specify the job repeat interval attribute (**inline**)
    - Setting an existing **Schedule** (interval-based or event-based)
  - **Destination**: where to run the job
    - Locally: as a database program unit or external executable
    - In a single named destination object
    - In a named destination group (list of remote locations)
  - **Credential**: run the job as what user
    - The owner
    - Specific credential

Creating a Time-Based Job

- Use the **DBMS_SCHEDULER.CREATE_JOB**:

```
DBMS_SCHEDULER.CREATE_JOB(
 JOB_NAME => 'SEND_SMS_JOB',
 JOB_TYPE => 'STORED_PROCEDURE',
 JOB_ACTION => 'SEND_SMS_CAMPAIGN',
 START_DATE => '28-JUNE-22 07.00.00 PM AUSTRALIA/SYDNEY',
 REPEAT_INTERVAL => 'FREQ=MONTHLY;INTERVAL=1',
 END_DATE => '30-DEC-22 07.00.00 PM AUSTRALIA/SYDNEY',
 COMMENTS => 'Monthly SMS campaign')
```

Parameter	Description
JOB_TYPE	**PLSQL_BLOCK**, **STORED_PROCEDURE**, **EXECUTABLE**
JOB_ACTION	specifies the exact procedure, command, or script that the job will execute.
START_DATE	These parameters specify the date that a new job should start and end
END_DATE	

Setting Repeat Intervals

- Set **REPEAT_INTERVAL** acceptable attributes:

Attribute	Possible Values
FREQ	**YEARLY, MONTHLY, WEEKLY, DAILY, HOURLY, MINUTELY, SECONDLY**
INTERVAL	A positive integer representing how often the recurrence repeats (default 1)
BYMONTH	Specifies which month or months you want the job to execute (numeric or three-letter abbreviations)
BYYEARDAY	Specifies the day of the year as a number. Valid values are 1 to 366.
BYDAY	Specifies the day of the month as a number. Valid values are 1 to 31.
BYHOUR, BYMINUTE, BYSECOND	

Repeat Interval Examples

Example	Meaning
`FREQ=DAILY;INTERVAL=10`	Every 10 days starting from the `START_DATE`
`FREQ=HOURLY;INTERVAL=2`	Every 2 hours starting from the `START_DATE`
`FREQ=WEEKLY;BYDAY=FRI`	Every Friday at midnight

`FREQ=WEEKLY;INTERVAL=2;`
`BYDAY=FRI`
	Every other Friday

`FREQ=MONTHLY;BYMONTHDAY=-1`	The last day of every month

`FREQ=YEARLY;BYMONTH=DEC;`
`BYMONTHDAY=31`
	On the 31st of December

`FREQ=MONTHLY;BYDAY=2FRI`	Every 2nd Friday of each month

- **Reference**: Refer to PL/SQL Packages and Types Reference, Table **-** Values for **`REPEAT_INTERVAL`**

Altering Jobs

- All job attributes can be changed (except job name)
- Set the **SET_ATTRIBUTE**, **SET_ATTRIBUTE_NULL**, or **SET_JOB_ATTRIBUTES** procedures in the **DBMS_SCHEDULER** package
- Example:

```
BEGIN
 DBMS_SCHEDULER.SET_ATTRIBUTE (
  NAME => 'SEND_SMS_JOB',
  ATTRIBUTE => 'REPEAT_INTERVAL',
  VALUE => 'FREQ=WEEKLY; BYDAY=FRI');
END;
/
```

Enabling, Disabling, and Removing Jobs

- To enable a job:

```
DBMS_SCHEDULER.ENABLE ('SEND_SMS_JOB')
```

- To disable a job:

```
DBMS_SCHEDULER.DISABLE ('SEND_SMS_JOB')
```

- To remove a job:

```
DBMS_SCHEDULER.DROP ('SEND_SMS_JOB')
```

Running and Stopping Jobs

- To manually run a job:

```
DBMS_SCHEDULER.RUN_JOB (
 JOB_NAME IN VARCHAR2,
 USE_CURRENT_SESSION IN BOOLEAN DEFAULT TRUE);
```

- Example:

```
DBMS_SCHEDULER.RUN_JOB ('SEND_SMS_JOB', FALSE)
```

- To stop a running job:

```
DBMS_SCHEDULER.STOP_JOB ('SEND_SMS_JOB')
```

Managing Programs

- A Program is a Scheduler object which represents a task or a collection of tasks that can be run by a job. Useful when we have multiple jobs running the same tasks.
- The procedures to manage programs are as follows:

Procedure	Description
`CREATE_PROGRAM`	create a program
`SET_ATTRIBUTE`	alter a program
`DROP_PROGRAM`	drop a program
`DISABLE`	disable a program
`ENABLE`	enable a program

Using Programs Example

```
DBMS_SCHEDULER.CREATE_PROGRAM(
 PROGRAM_NAME => 'SEND_SMS_PROG',
 PROGRAM_ACTION => 'MARKETING.SEND_SMS_CAMPAIGN',
 PROGRAM_TYPE => 'STORED_PROCEDURE',
 ENABLED => TRUE);
```

```
DBMS_SCHEDULER.CREATE_JOB(
 JOB_NAME => 'SEND_SMS_JOB',
 PROGRAM_NAME => 'SEND_SMS_PROG',
 REPEAT_INTERVAL => 'FREQ=WEEKLY; BYDAY=FRI',
 ENABLED => TRUE);
```

Managing Schedules

- A Schedule is a Scheduler object which defines when a job should be run. Normally useful when we have multiple jobs running on the same schedule.
- The procedures to manage programs are as follows:

Procedure	Description
`CREATE_SCHEDULE`	create a schedule
`SET_ATTRIBUTE`	alter a schedule
`DROP_SCHEDULE`	drop a schedule

- Created schedules are granted to **`PUBLIC`** (all users can access it)

Using Schedule Example

```
BEGIN
 DBMS_SCHEDULER.CREATE_SCHEDULE (
  SCHEDULE_NAME => 'CAMPAIGN_SCHEDULE',
  START_DATE => SYSTIMESTAMP,
  END_DATE => SYSTIMESTAMP + INTERVAL '30' DAY,
  REPEAT_INTERVAL => 'FREQ=WEEKLY; INTERVAL=1',
  COMMENTS => 'Every week starting from today and ends in 30 days');
END;
/
```

```
DBMS_SCHEDULER.CREATE_JOB(
 JOB_NAME => 'SEND_SMS_JOB',
 PROGRAM_NAME => 'SEND_SMS_PROG',
 SCHEDULE_NAME => 'CAMPAIGN_SCHEDULE',
 ENABLED => TRUE);
```

Monitoring Scheduler Jobs by Querying Dict. Views

- Scheduler job log is implemented as the following data dictionary views:
  - **`*_SCHEDULER_JOB_LOG`** (parent)
  - **`*_SCHEDULER_JOB_RUN_DETAILS`** (child linked by **`LOG_ID`**)

```
SELECT TO_CHAR(LOG_DATE, 'DD-MON-YY HH24:MI:SS') TIMESTAMP,
JOB_NAME, OPERATION, STATUS FROM USER_SCHEDULER_JOB_LOG
WHERE JOB_NAME LIKE 'ORA$%' ORDER BY LOG_DATE DESC;

TIMESTAMP                 JOB_NAME            OPERATION  STATUS
------------------------- -------------------- ---------- ---------
24-MAY-23 20:00:30        ORA$AT_OS_OPT_SY_1   RUN        SUCCEEDED
24-MAY-23 20:00:30        ORA$AT_OS_OPT_SY_1   COMPLETED
24-MAY-23 20:00:30        ORA$AT_OS_OPT_SY_1   DROP
```

Obtain Information About Scheduler Objects

Dictionary View	Description
`*_SCHEDULER_JOBS`	Retrieves information about the Scheduler jobs
`*_SCHEDULER_SCHEDULES`	Retrieves information about the running Scheduler Schedules
`*_SCHEDULER_PROGRAMS`	Retrieves information about the running Scheduler Programs
`*_SCHEDULER_JOB_LOG`	Retrieves log information for the Scheduler jobs
`*_SCHEDULER_JOB_RUN_DETAILS`	Retrieves log run details for the Scheduler jobs
`*_SCHEDULER_RUNNING_JOBS`	Retrieves information about the running Scheduler jobs

More About the Scheduler

Scheduler Object	Meaning
Job Class	Helps you prioritize jobs by allocating resources differently among the various jobs.
Windows	Windows enable the automatic changing of resource plans based on a schedule.
Window Groups	A window group is a collection of windows. When a window group is assigned to the SCHEDULE_NAME attribute, the job runs in all the windows in the window group.
Event-Based Scheduling	Jobs can be triggered based on events. An application can notify the Scheduler to start a job by enqueuing a message onto an Oracle Streams AQ queue.
Chain	A chain is a named series of programs that are linked together for a combined objective.
Remote External Jobs	Used to create an external job that runs on a remote host.

Summary

In this lecture, you should have learnt how to perform the following:

- Describe scheduler job attributes
- Create and manage time-based jobs
- Manage scheduler programs
- Manage scheduler schedules
- Monitor Scheduler jobs by querying dictionary views