Oracle 11g DBA Fundamentals Overview

Lesson 12: Automating Tasks with the Scheduler

Objectives

After completing this lesson, you should be able to:

- Simplify management tasks by using the Scheduler
- Create a job, program, and schedule
- Monitor job execution
- Use a time-based or event-based schedule for executing Scheduler jobs

Simplifying Management Tasks

Performing a series of month-end tasks on the last day of each month

Running a dequeue procedure as soon as a message is enqueued

Replicating table data via materialized view refreshes

Running a daily job to back up database

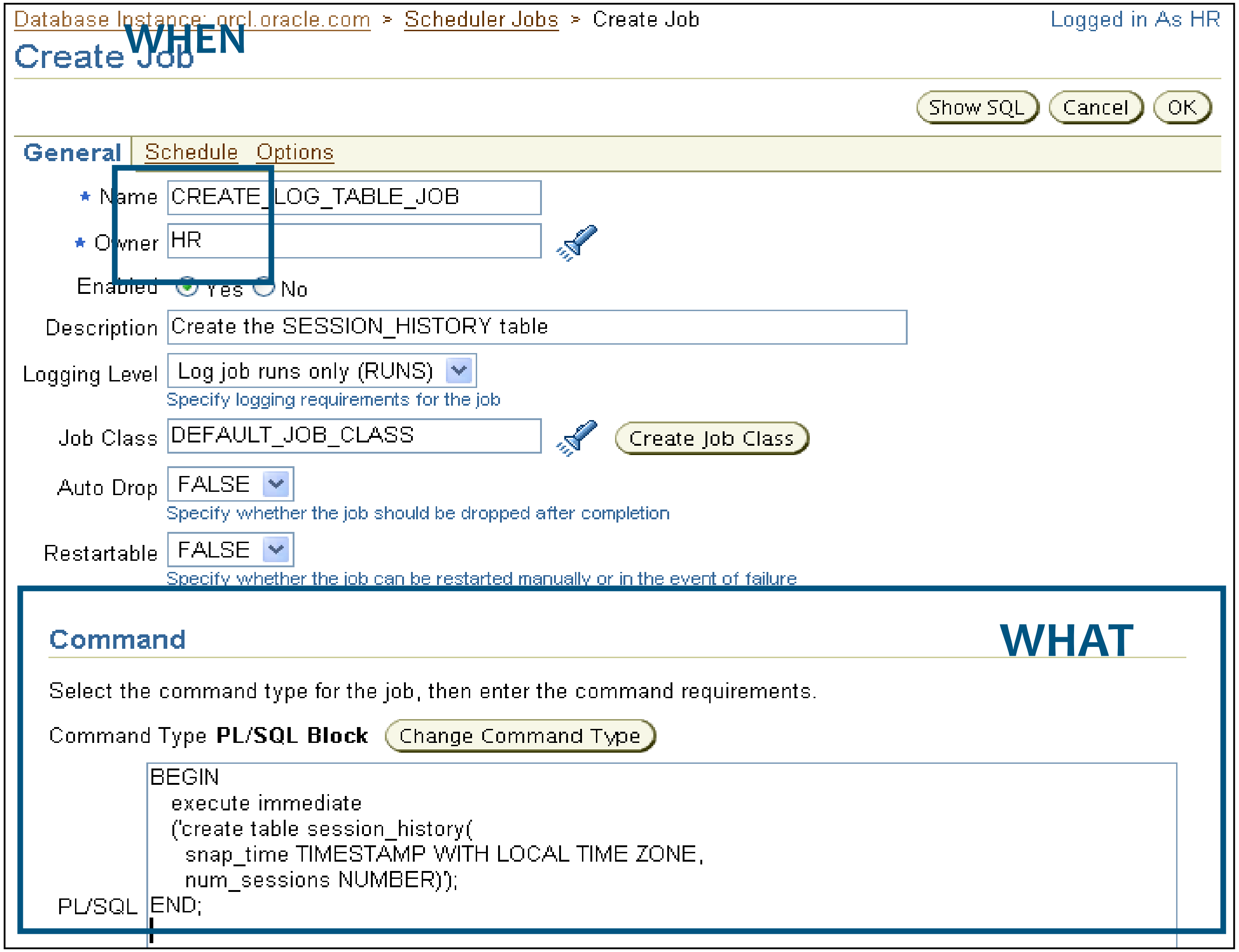
Computing table and index statistics twice a day

Starting the batch load as soon as the file arrives on the file system

Generating an hourly report on invalid server access attempts

Rebuilding an index when finished rebuilding the current index

A Simple Job



WHEN

Create Job

* Name CREATE_LOG_TABLE_JOB

* Owner HR

Enabled ● Yes ○ No

Description Create the SESSION_HISTORY table

Logging Level Log job runs only (RUNS)

Specify logging requirements for the job

Job Class DEFAULT_JOB_CLASS

Auto Drop FALSE

Specify whether the job should be dropped after completion

Restartable FALSE

Specify whether the job can be restarted manually or in the event of failure

Command

WHAT

Select the command type for the job, then enter the command requirements.

Command Type **PL/SQL Block**

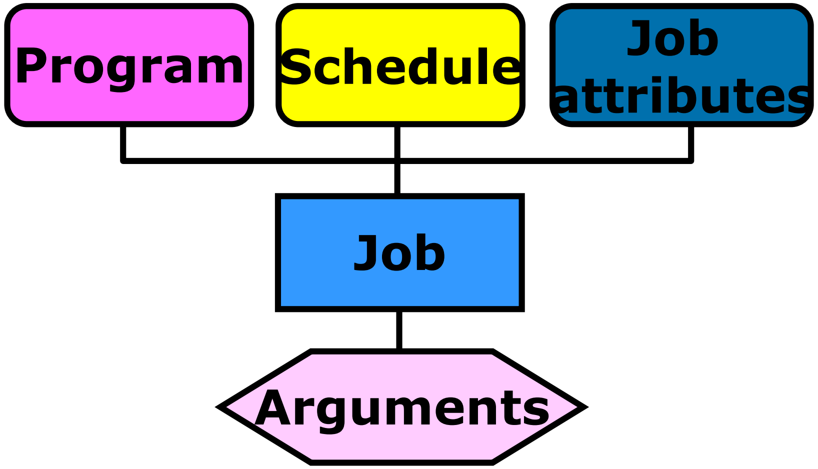
PL/SQL

```
BEGIN
  execute immediate
  ('create table session_history(
    snap_time TIMESTAMP WITH LOCAL TIME ZONE,
    num_sessions NUMBER)');
END;
```

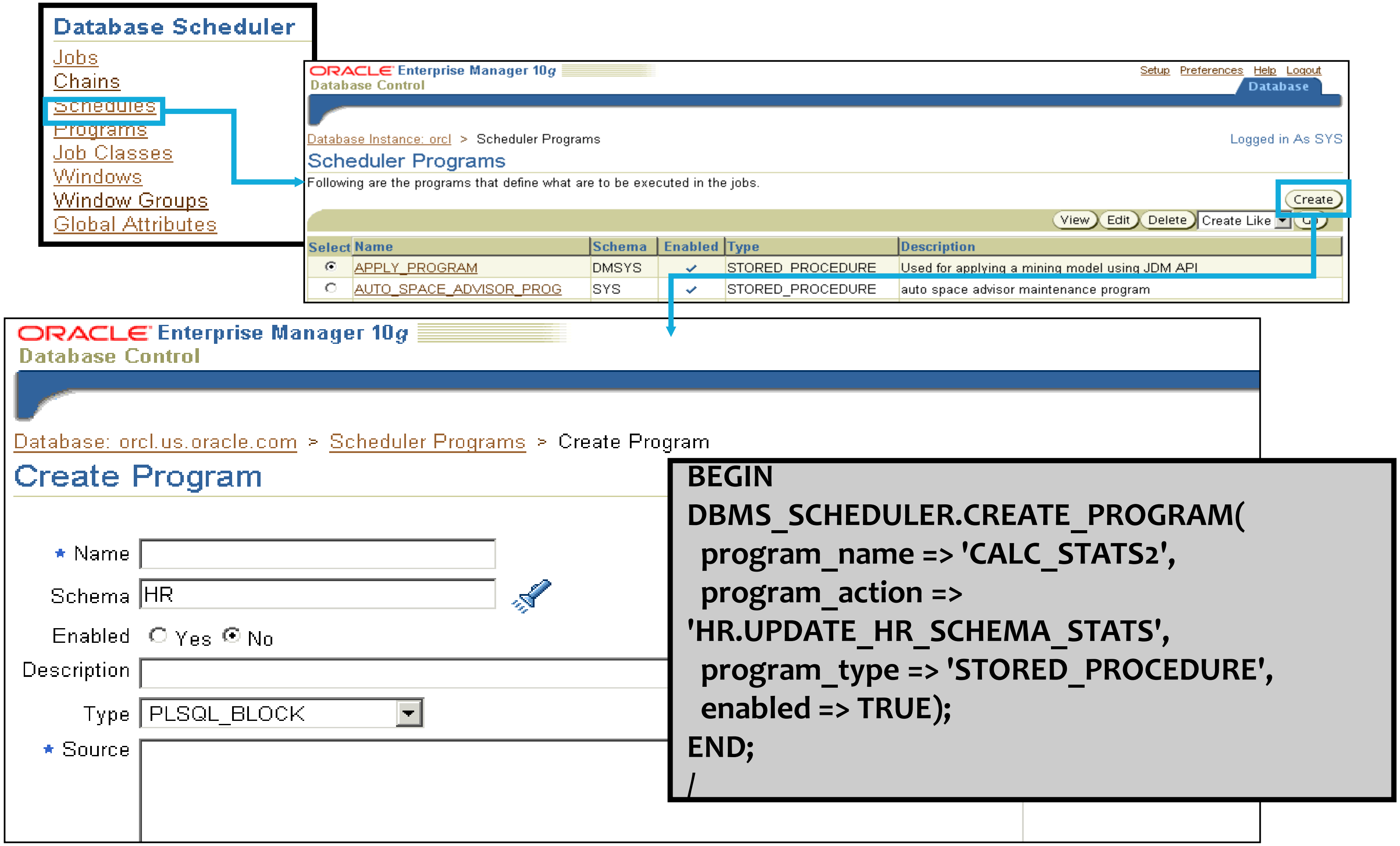
Key Components and Steps

To simplify management tasks with the Scheduler, perform the following steps:

1. Create a program.
2. Create and use a schedule.
3. Create and submit a job.
4. Monitor a job.



1. Creating a Program



Database Instance: orcl > Scheduler Programs

Logged in As SYS

Scheduler Programs

Following are the programs that define what are to be executed in the jobs.

Select	Name	Schema	Enabled	Type	Description
●	APPLY_PROGRAM	DMSYS	✓	STORED_PROCEDURE	Used for applying a mining model using JDM API
○	AUTO_SPACE_ADVISOR_PROG	SYS	✓	STORED_PROCEDURE	auto space advisor maintenance program

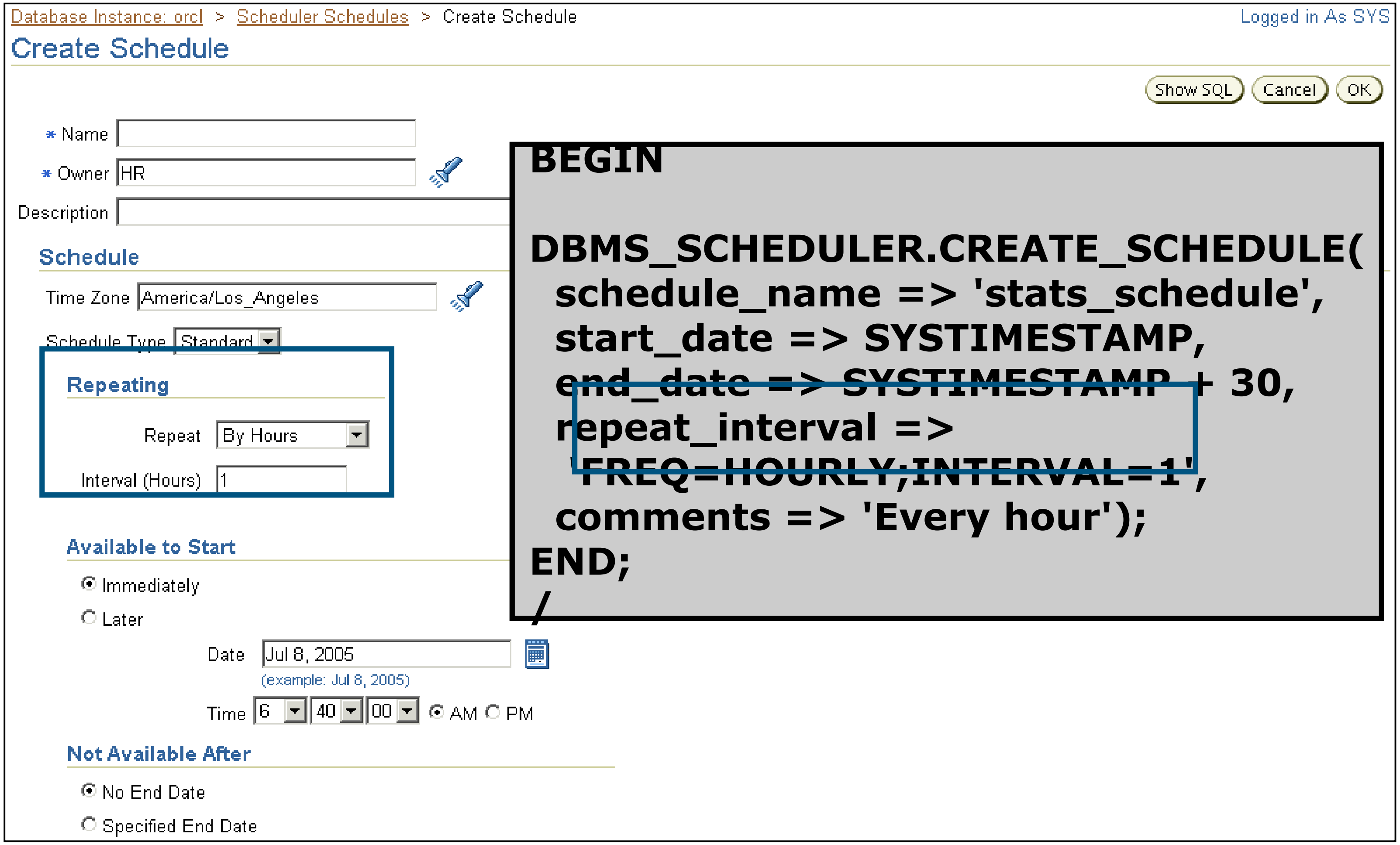
Database: orcl.us.oracle.com > Scheduler Programs > Create Program

Create Program

* Name	
Schema	HR
Enabled	○ Yes ● No
Description	
Type	PLSQL_BLOCK
* Source	

```
BEGIN
DBMS_SCHEDULER.CREATE_PROGRAM(
 program_name => 'CALC_STATS2',
 program_action =>
'HR.UPDATE_HR_SCHEMA_STATS',
 program_type => 'STORED_PROCEDURE',
 enabled => TRUE);
END;
/
```

2. Creating and Using Schedules



Database Instance: orcl > Scheduler Schedules > Create Schedule
Logged in As SYS
Show SQL
Cancel
OK

Create Schedule

* Name

* Owner HR

Description

Schedule

Time Zone America/Los_Angeles

Schedule Type Standard

Repeating

Repeat By Hours

Interval (Hours) 1

Available to Start

● Immediately

○ Later

Date Jul 8, 2005
(example: Jul 8, 2005)

Time 6 40 00 ● AM ○ PM

Not Available After

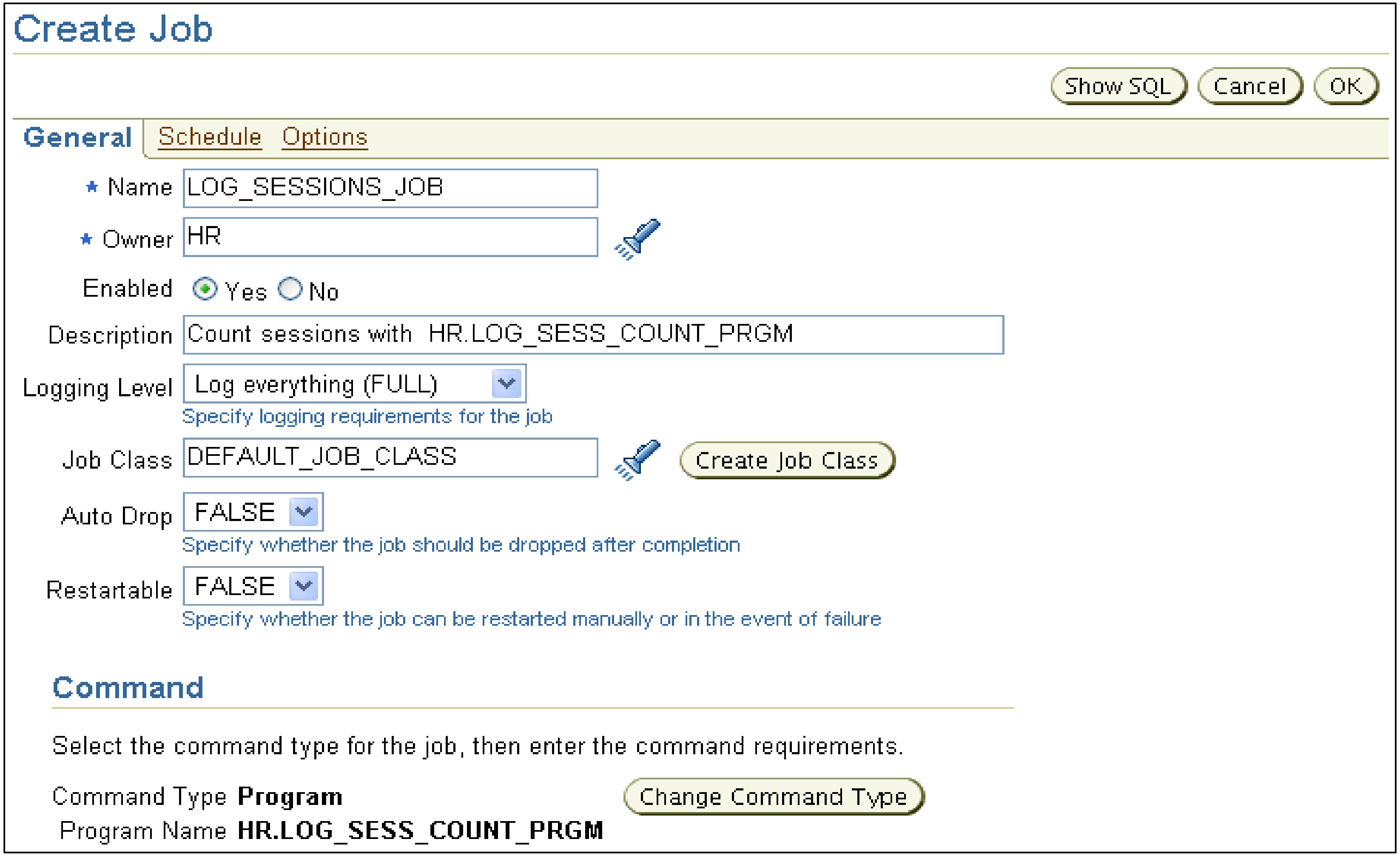
● No End Date

○ Specified End Date

```
BEGIN

DBMS_SCHEDULER.CREATE_SCHEDULE(
  schedule_name => 'stats_schedule',
  start_date => SYSTIMESTAMP,
  end_date => SYSTIMESTAMP + 30,
  repeat_interval =>
   'FREQ=HOURLY;INTERVAL=1',
  comments => 'Every hour');
END;
/
```

3. Creating and Running a Job



Create Job

* Name LOG_SESSIONS_JOB

* Owner HR

Enabled ● Yes ○ No

Description Count sessions with HR.LOG_SESS_COUNT_PRGM

Logging Level Log everything (FULL)
Specify logging requirements for the job

Job Class DEFAULT_JOB_CLASS

Auto Drop FALSE
Specify whether the job should be dropped after completion

Restartable FALSE
Specify whether the job can be restarted manually or in the event of failure

Command

Select the command type for the job, then enter the command requirements.

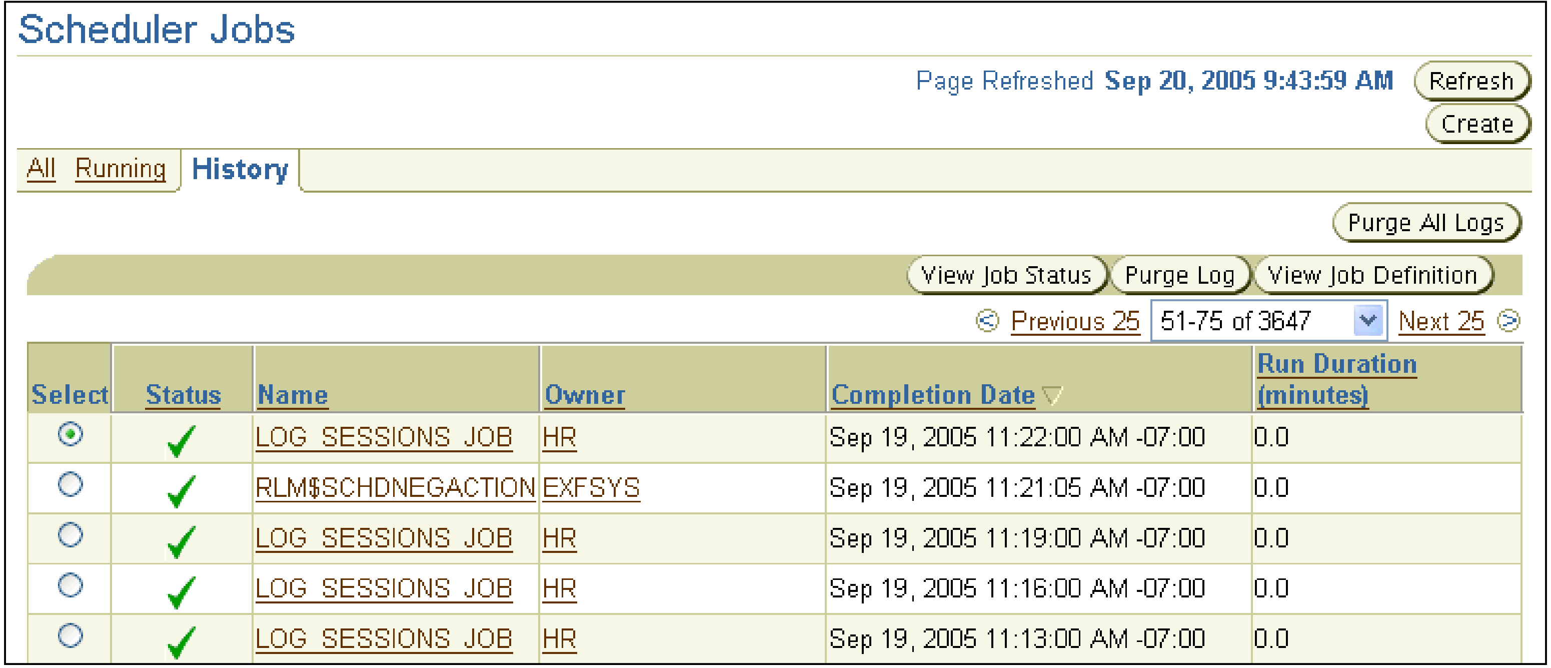
Command Type **Program**

Program Name **HR.LOG_SESS_COUNT_PRGM**

4. Monitoring a Job

```
SELECT job_name, status, error#, run_duration
FROM USER_SCHEDULER_JOB_RUN_DETAILS;

JOB_NAME          STATUS ERROR#  RUN_DURATION
----------------  ------ ------  ------------
GATHER_STATS_JOB  SUCCESS     0 +000 00:08:20
PART_EXCHANGE_JOB FAILURE  6576 +000 00:00:00
```



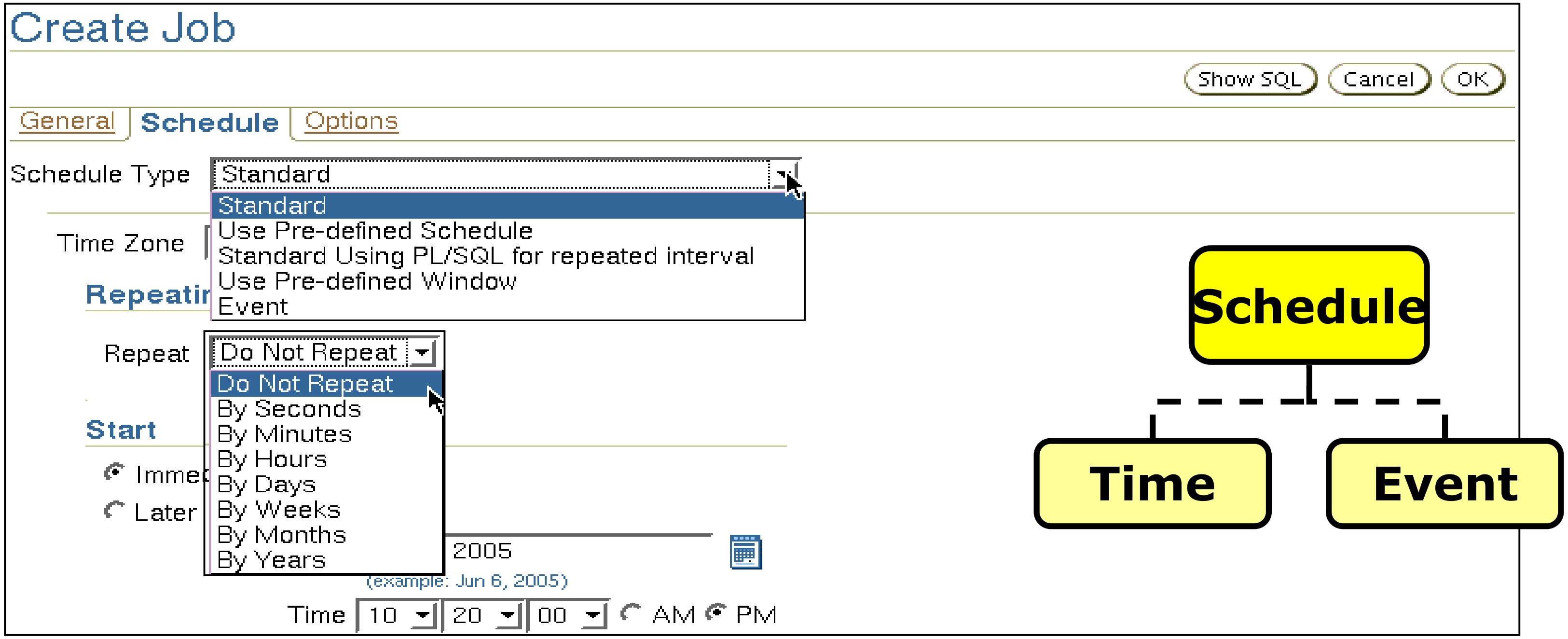
Scheduler Jobs

Page Refreshed **Sep 20, 2005 9:43:59 AM**

Select	Status	Name	Owner	Completion Date ▽	Run Duration (minutes)
◉	✓	LOG_SESSIONS_JOB	HR	Sep 19, 2005 11:22:00 AM -07:00	0.0
○	✓	RLM$SCHDNEGACTION	EXFSYS	Sep 19, 2005 11:21:05 AM -07:00	0.0
○	✓	LOG_SESSIONS_JOB	HR	Sep 19, 2005 11:19:00 AM -07:00	0.0
○	✓	LOG_SESSIONS_JOB	HR	Sep 19, 2005 11:16:00 AM -07:00	0.0
○	✓	LOG_SESSIONS_JOB	HR	Sep 19, 2005 11:13:00 AM -07:00	0.0

Using a Time-Based or Event-Based Schedule

Key Comp.
& Steps
> Schedules
Job Chains
Adv.Concepts



Create Job

Schedule Type: Standard

Time Zone

Repeatir

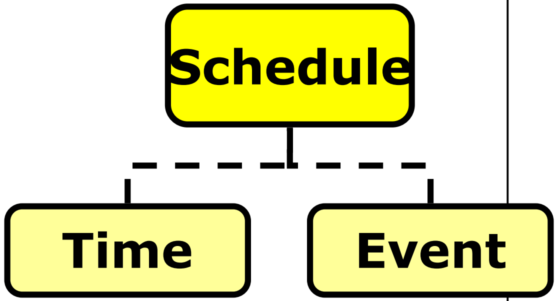
Repeat: Do Not Repeat

Start

● Immec
○ Later

2005
(example: Jun 6, 2005)

Time 10 20 00 ○ AM ● PM



Creating a Time-Based Job

Example:

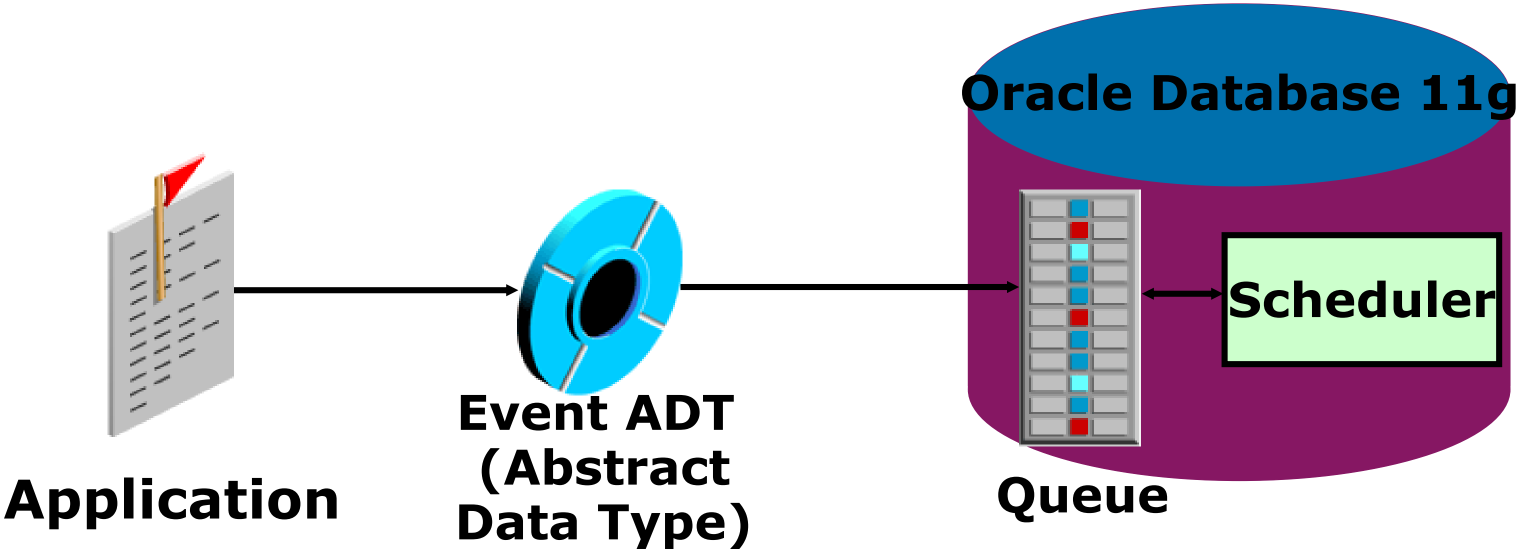
- Create a job that calls a backup script every night at 11:00, starting tonight.

```
BEGIN
 DBMS_SCHEDULER.CREATE_JOB(
  job_name=>'HR.DO_BACKUP',
  job_type => 'EXECUTABLE',
  job_action =>
'/home/usr/dba/rman/nightly_incr.sh',
  start_date=> SYSDATE,
  repeat_interval=>'FREQ=DAILY;BYHOUR=23',
                /* next night at 11:00 PM */
  comments => 'Nightly incremental backups');
END;
/
```

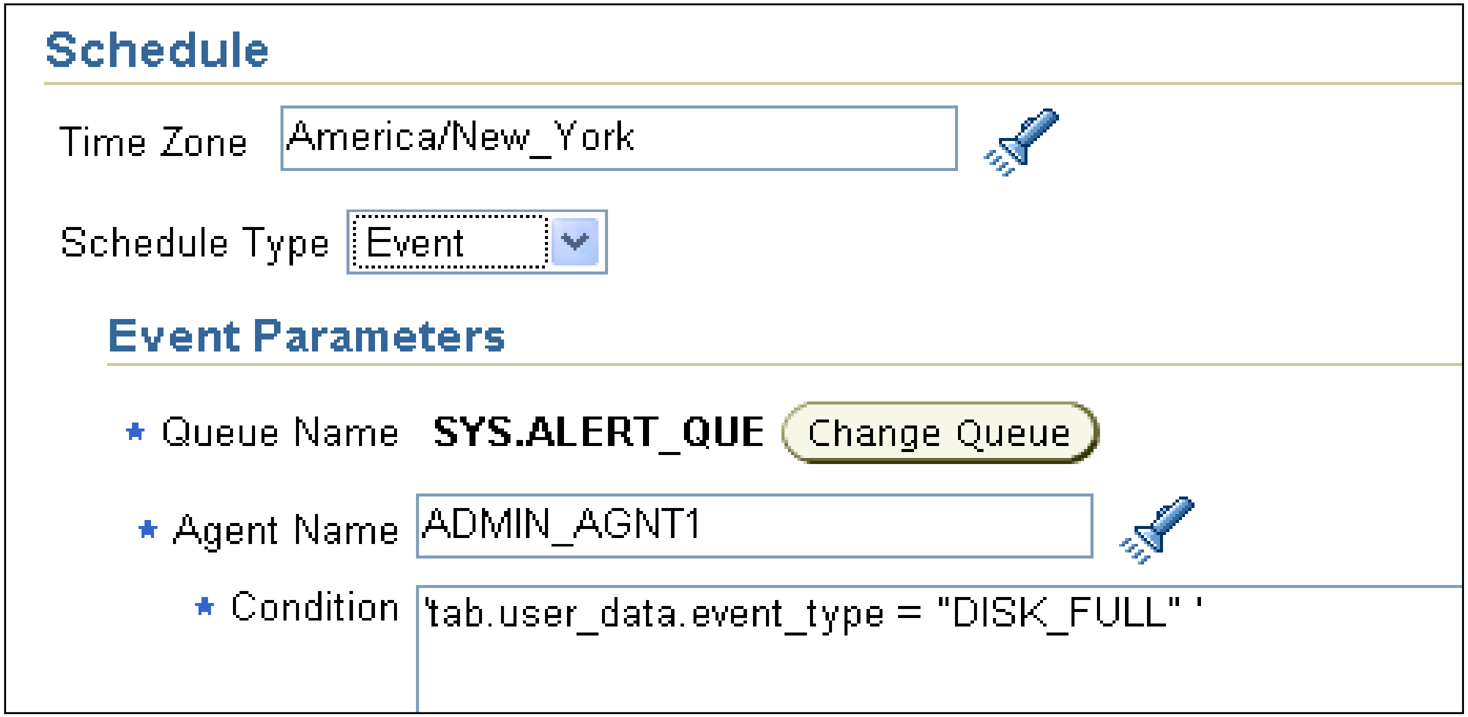
Creating an Event-Based Schedule

To create an event-based job, you must set:

- A queue specification (where your application enqueues messages to start a job)
- An event condition (same syntax as an Oracle Streams AQ rule condition) that if TRUE starts the job



Creating Event-Based Schedules with Enterprise Manager



Schedule

Time Zone: America/New_York

Schedule Type: Event

Event Parameters

* Queue Name: **SYS.ALERT_QUE**

* Agent Name: ADMIN_AGNT1

* Condition: 'tab.user_data.event_type = "DISK_FULL" '

Creating an Event-Based Job

Example:Create a job that runs if a batch load data file arrives on the file system before 9:00 a.m.

```
BEGIN
 DBMS_SCHEDULER.CREATE_JOB(
  job_name=>'ADMIN.PERFORM_DATA_LOAD',
  job_type => 'EXECUTABLE',
  job_action => '/home/usr/dba/rman/report_failure.sh',
  start_date => SYSTIMESTAMP,
   event_condition => 'tab.user_data.object_owner =
    ''HR'' and tab.user_data.object_name = ''DATA.TXT''
    and tab.user_data.event_type = ''FILE_ARRIVAL''
    and tab.user_data.event_timestamp < 9 ',
    queue_spec => 'HR.LOAD_JOB_EVENT_Q');
END;
```

Event-Based Scheduling

Event types:

- User- or application-generated events
- Scheduler-generated events
- Events raised by Scheduler jobs:
- JOB_START
  - JOB_SCH_LIM_REACHED
- JOB_SUCCEEDED
- JOB_DISABLED
- JOB_FAILED
  - JOB_CHAIN_STALLED
- JOB_BROKEN
  - JOB_ALL_EVENTS
- JOB_COMPLETED
- JOB_STOPPED
- Example of raising a

```
DBMS_SCHEDULER.SET_ATTRIBUTE('hr.do_bac
 'raise_events', DBMS_SCHEDULER.JOB_FAILE
```

SUMMARY

- In this lesson, you should have learned how to:
  - Simplify management tasks by using the Scheduler
  - Create a job, program, and schedule
  - Monitor job execution
  - Use a time-based or event-based schedule for executing Scheduler jobs