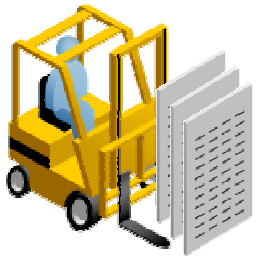
# 18

# Moving Data



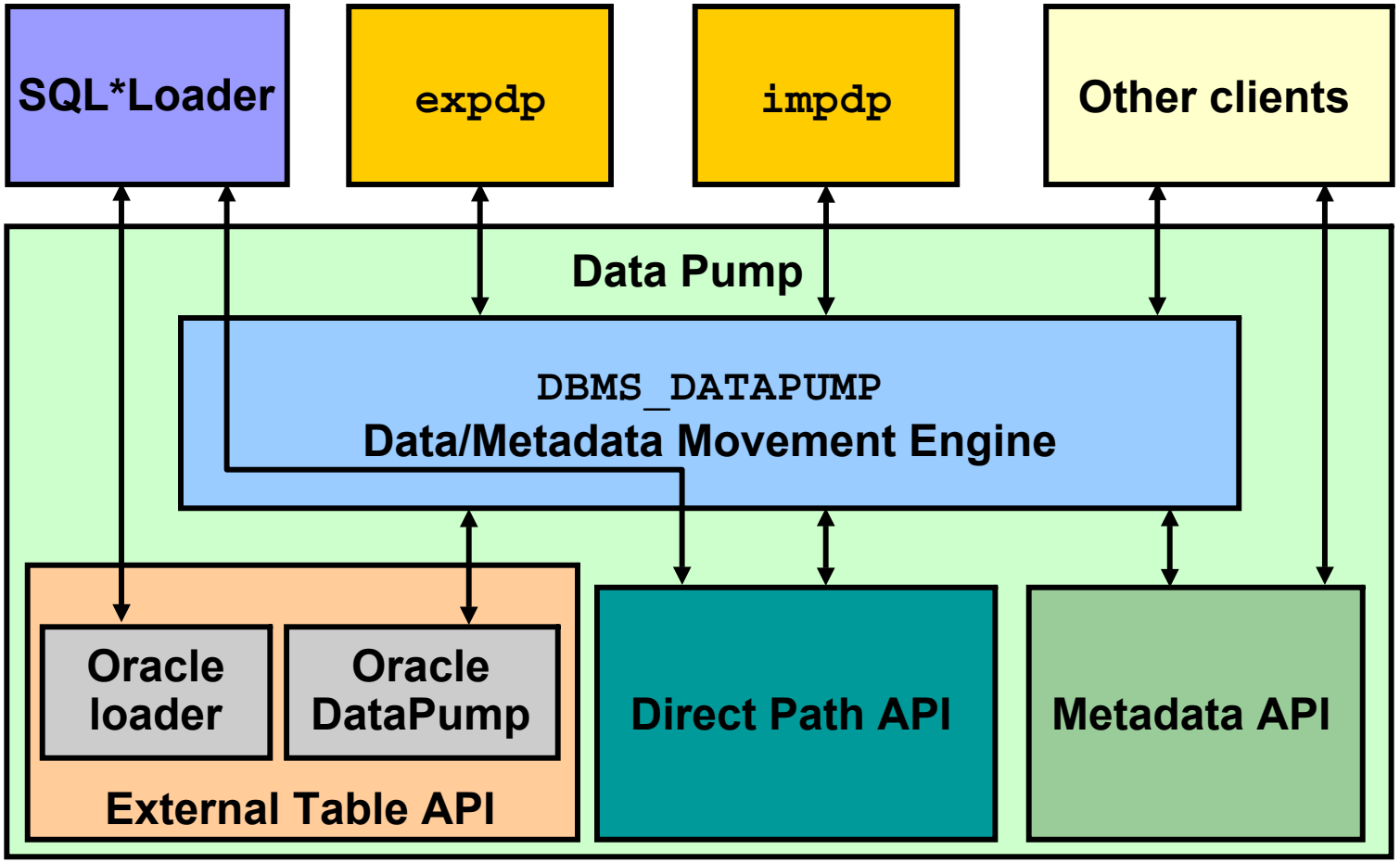
# Objectives

Directory Obj.
SQL*Loader
Data Pump
- Export
- Import
External Table

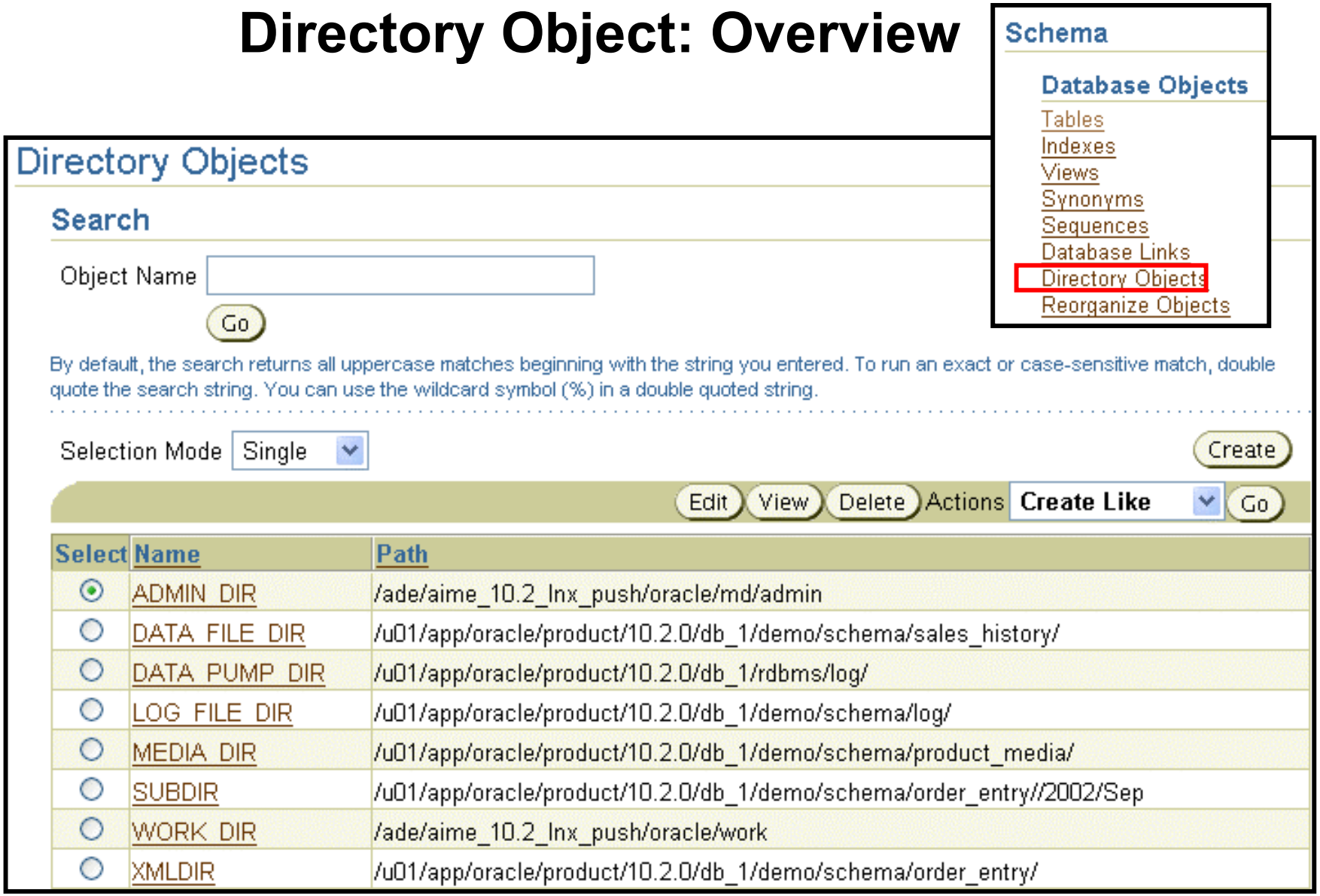
After completing this lesson, you should be able to do the following:

- Describe available ways for moving data
- Create and use directory objects
- Use SQL*Loader to load data from a non-Oracle database (or user files)
- Explain the general architecture of Data Pump
- Use Data Pump Export and Import to move data between Oracle databases
- Use external tables to move data via platform-independent files

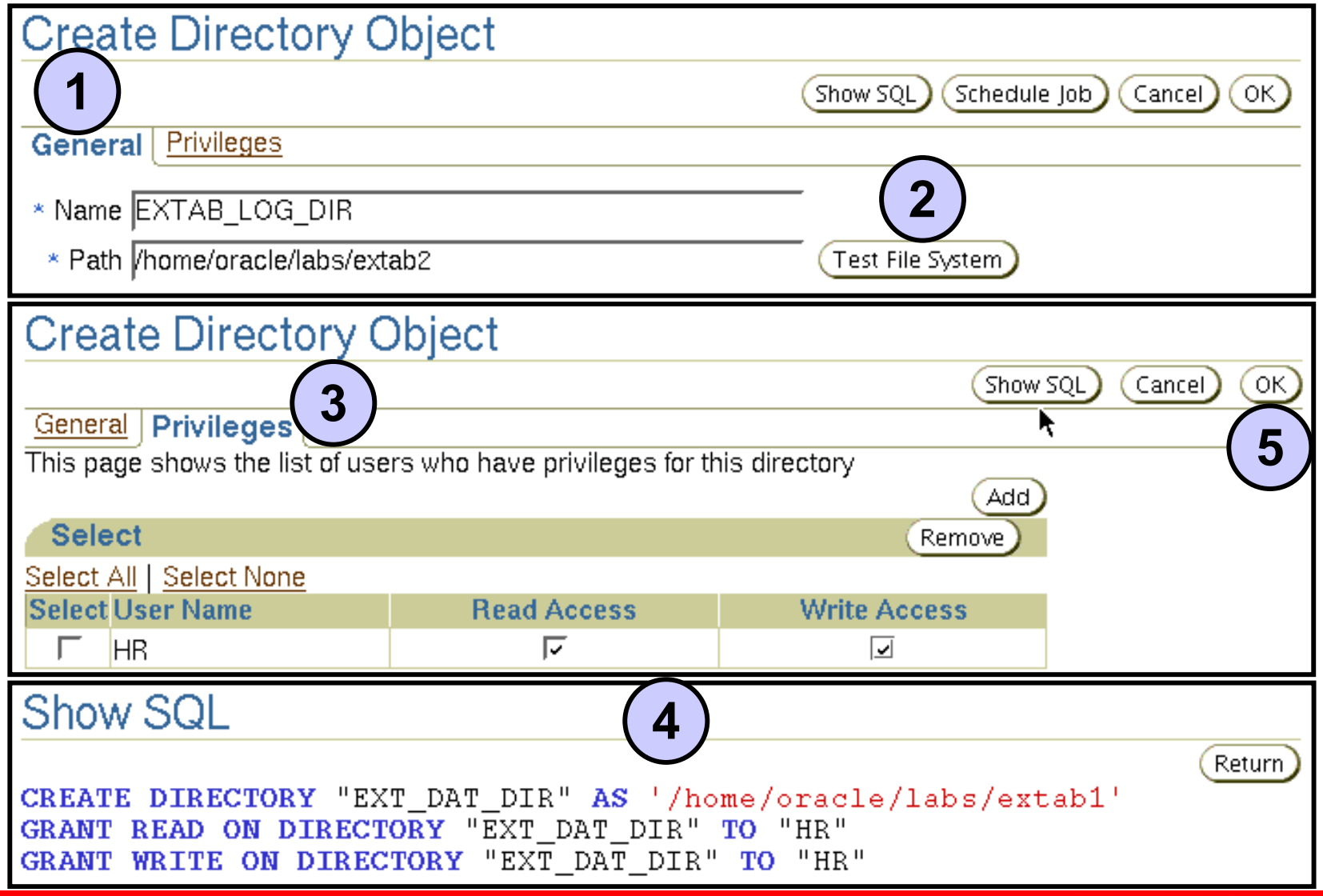
# Moving Data: General Architecture



# Directory Object: Overview

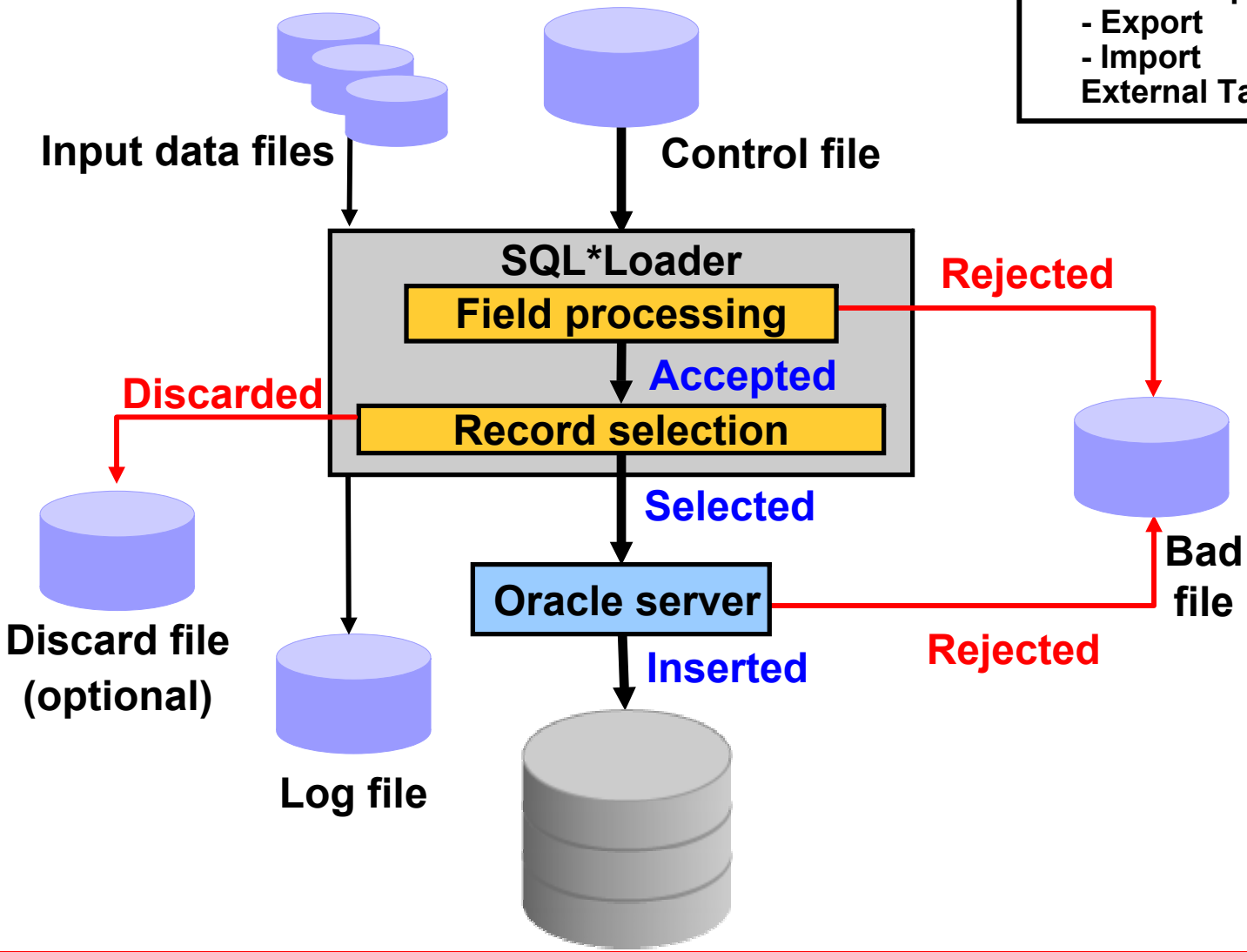


# Creating Directory Objects

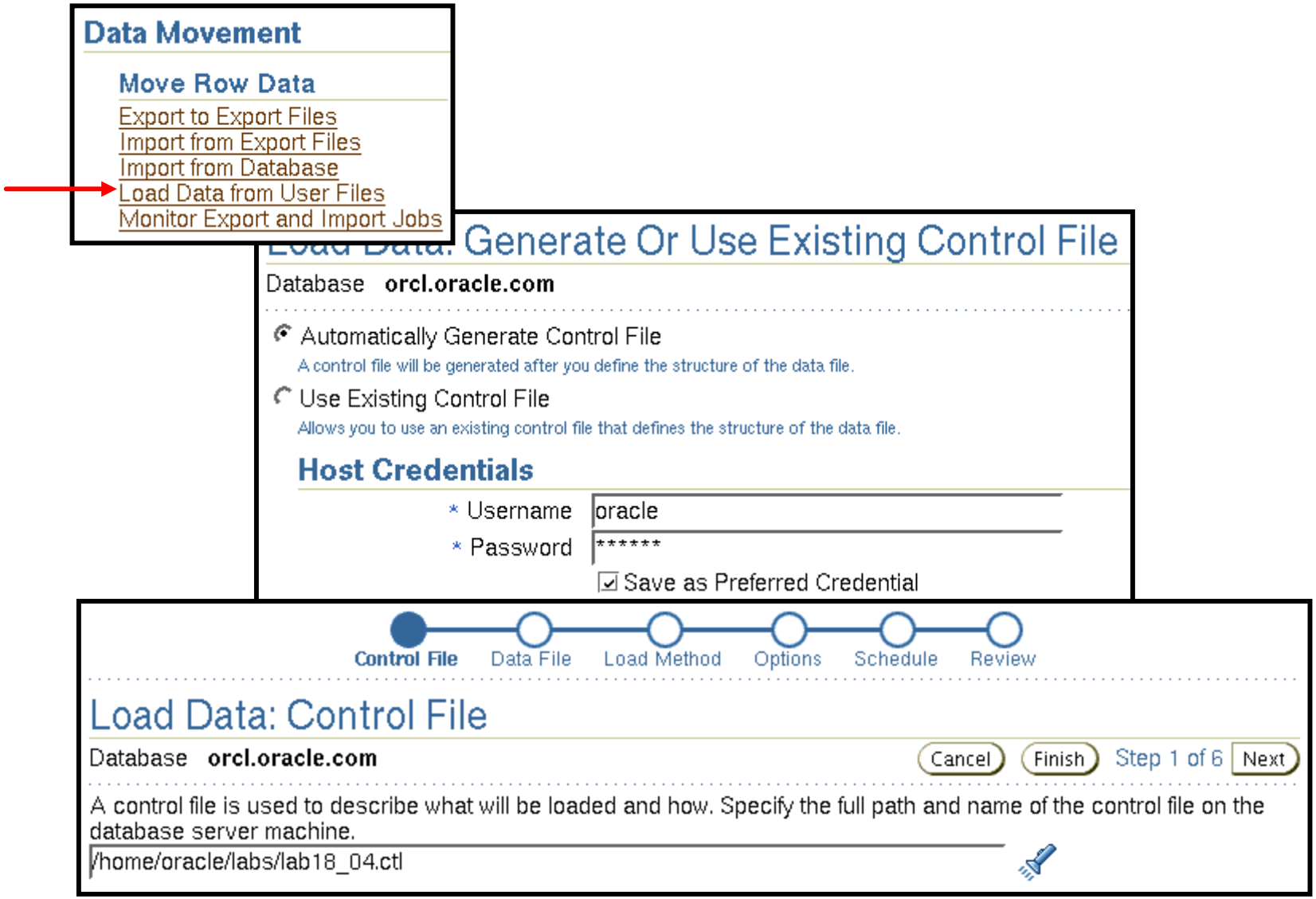


# SQL*Loader: Overview

- Directory Obj.
- > SQL*Loader
- Data Pump
  - Export
  - Import
- External Table



# Loading Data with SQL*Loader

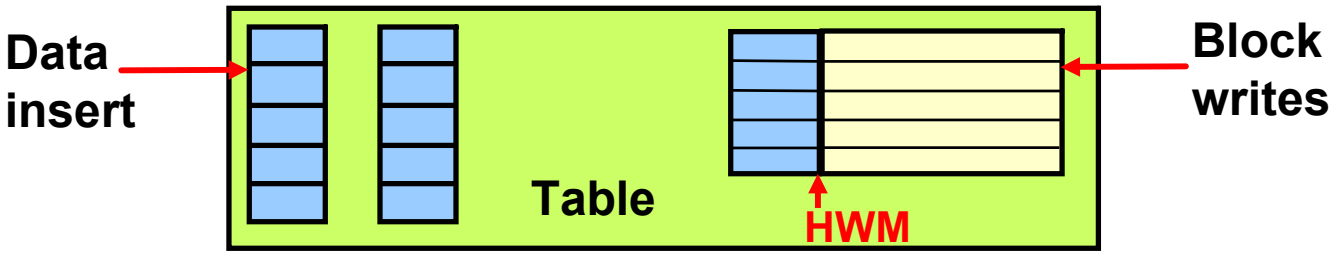


# SQL*Loader Control File

The SQL*Loader control file instructs SQL*Loader about:

- Location of the data to be loaded
- The data format
- Configuration details:
  - Memory management
  - Record rejection
  - Interrupted load handling details
- Data manipulation details

# Loading Methods




| Conventional Load | Direct Path Load |
|---|---|
| Uses `COMMIT` | Uses data saves (faster operation) |
| Always generates redo entries | Generates redo only under specific conditions |
| Enforces all constraints | Enforces only `PRIMARY KEY`, `UNIQUE`, and `NOT NULL` |
| Fires `INSERT` triggers | Does not fire `INSERT` triggers |
| Can load into clustered tables | Does not load into clusters |
| Allows other users to modify tables during load operation | Prevents other users from making changes to tables during load operation |

# Data Pump: Overview

Directory Obj.
SQL*Loader
> Data Pump
- Export
- Import
External Table

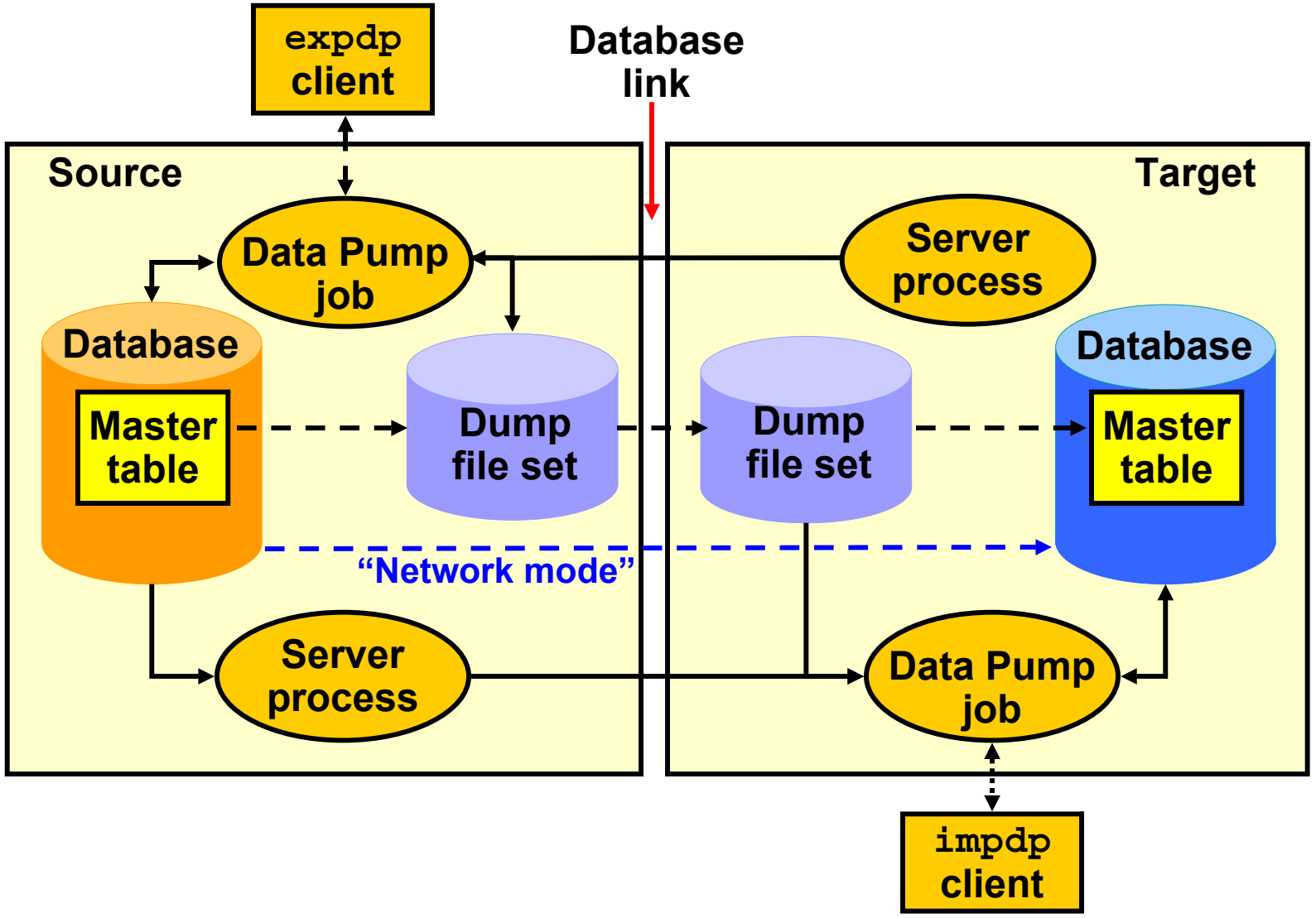
As a server-based facility for high-speed data and metadata movement, Data Pump:

- Is callable via `DBMS_DATAPUMP`
- Provides the following tools:
  - `expdp`
  - `impdp`
  - Web-based interface
- Provides data access methods:
  - Direct path
  - External tables
- Detaches from and reattaches to long-running jobs
- Restarts Data Pump jobs

# Data Pump: Benefits

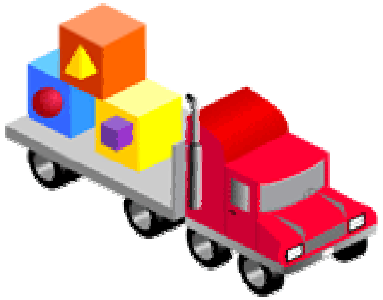
- Fine-grained object and data selection
- Explicit specification of database version
- Parallel execution
- Estimation of the export job space consumption
- Network mode in a distributed environment
- Remapping capabilities during import
- Data sampling and metadata compression

# Data Pump Export and Import: Overview



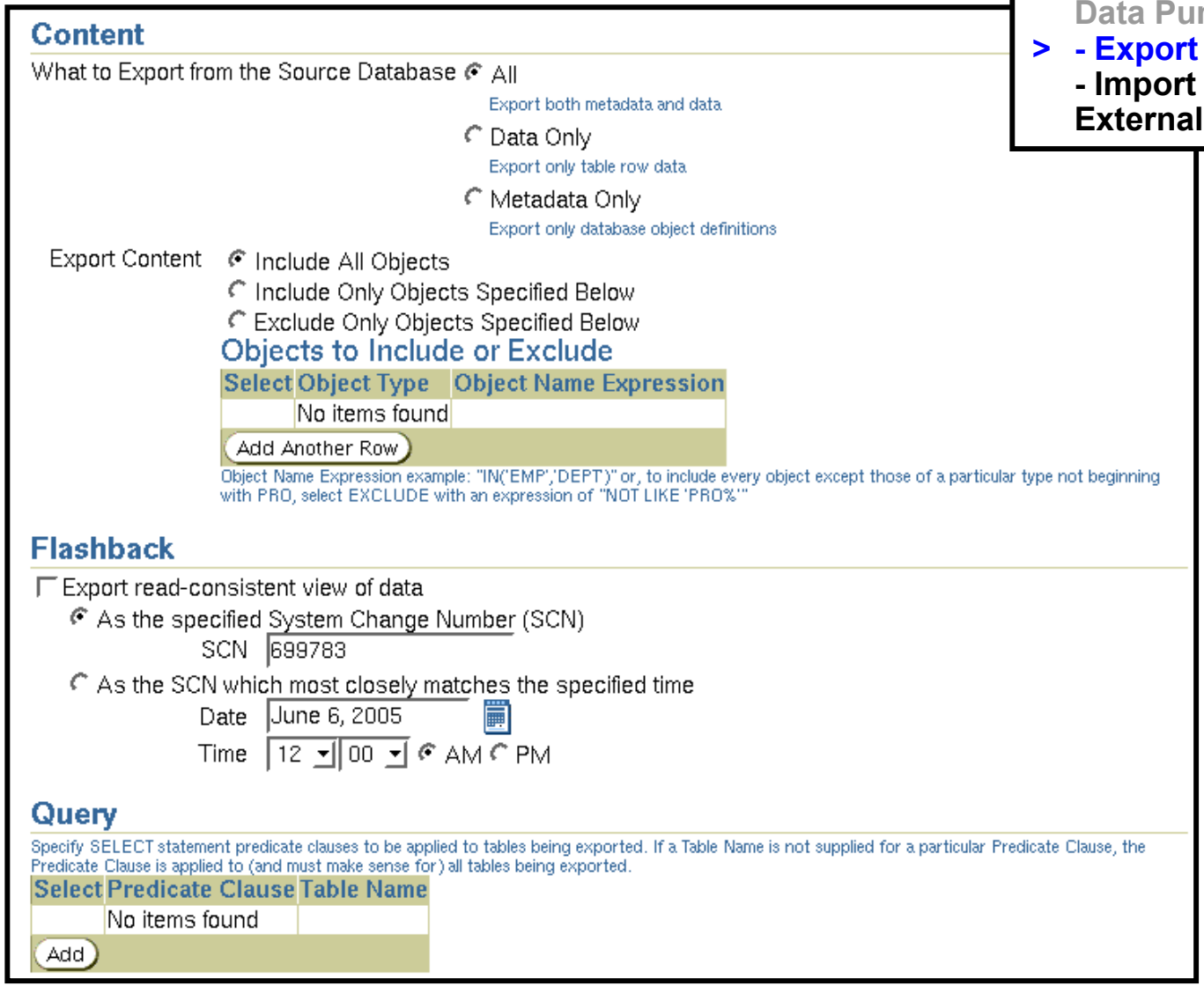
# Data Pump Utility: Interfaces and Modes

- Data Pump Export and Import interfaces:
  - Command line
  - Parameter file
  - Interactive command line
  - Database Control
- Data Pump Export and Import modes:
  - Full
  - Schema
  - Table
  - Tablespace
  - Transportable tablespace



# Fine-Grained Object Selection

- Directory Obj.
- SQL*Loader
- Data Pump
- > - Export
- - Import
- External Table



# Advanced Feature: Sampling

- Task: Create test data.
- Method: Specify a percentage of data to be sampled and unloaded from the source database.

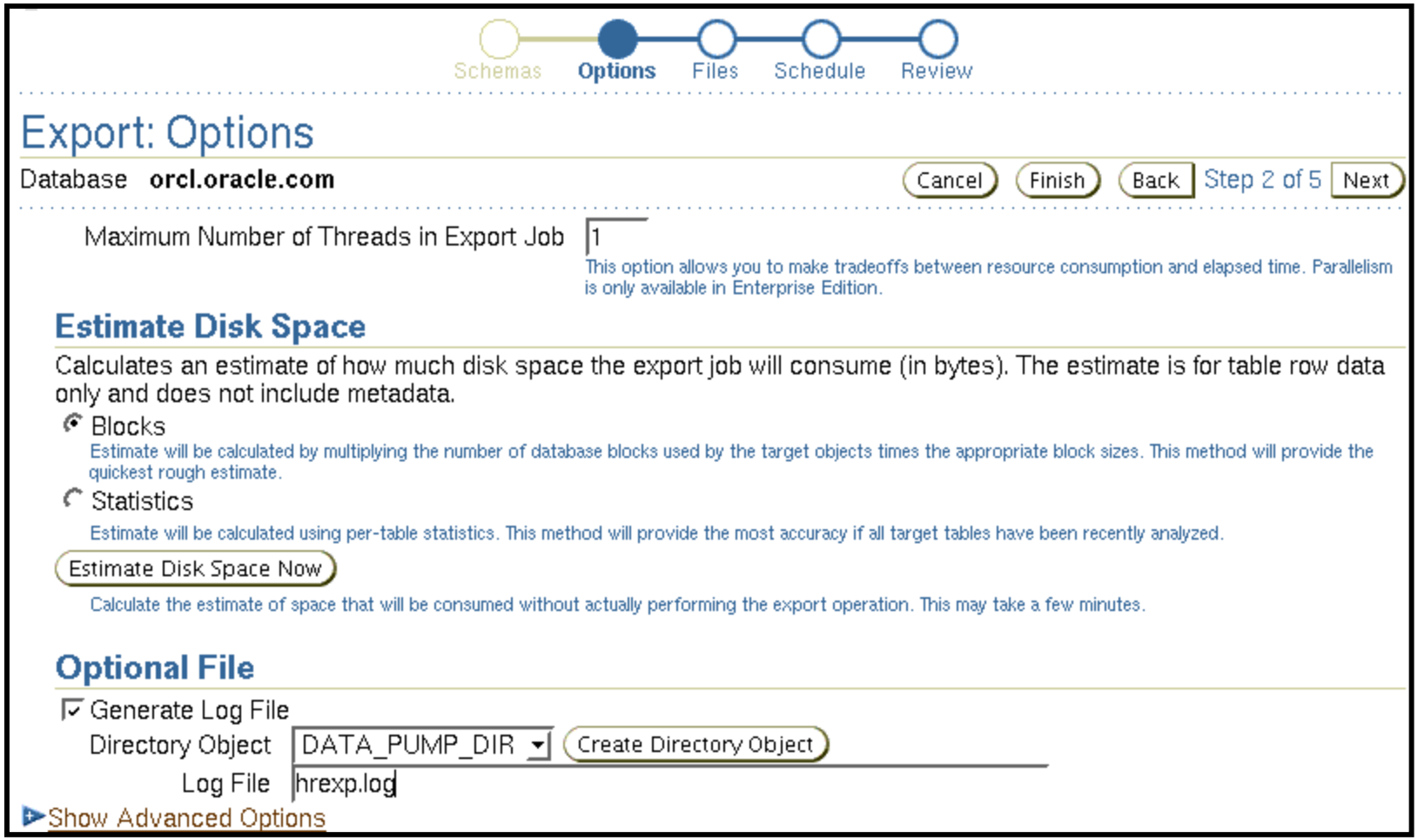
Example to unload 44% of the `HR.EMPLOYEES` table:

```
SAMPLE="HR"."EMPLOYEES":44
```

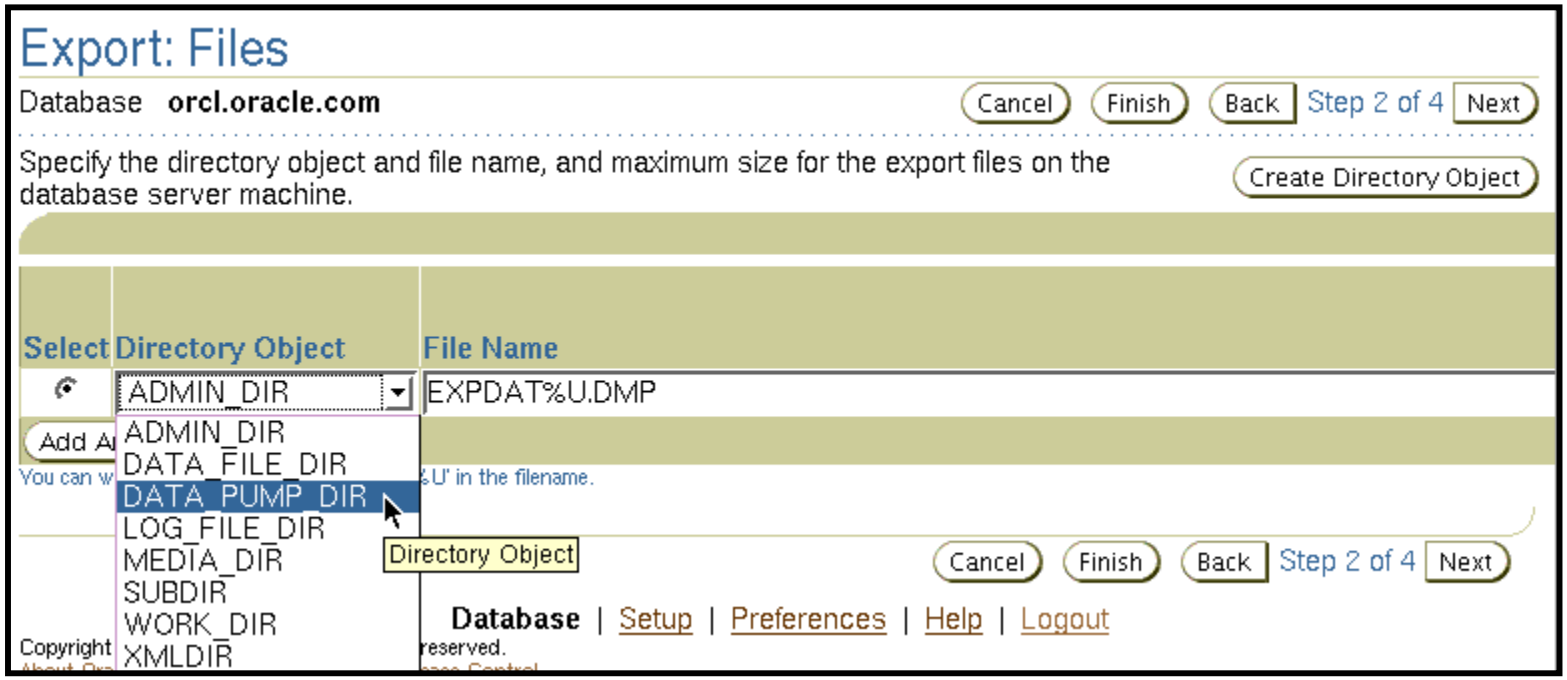
Example to unload 30% of the entire export job (because no table name is specified):

```
expdp hr/hr DIRECTORY=DATA_PUMP_DIR
DUMPFILE=sample1.dmp SAMPLE=30
```

# Export Options: Files



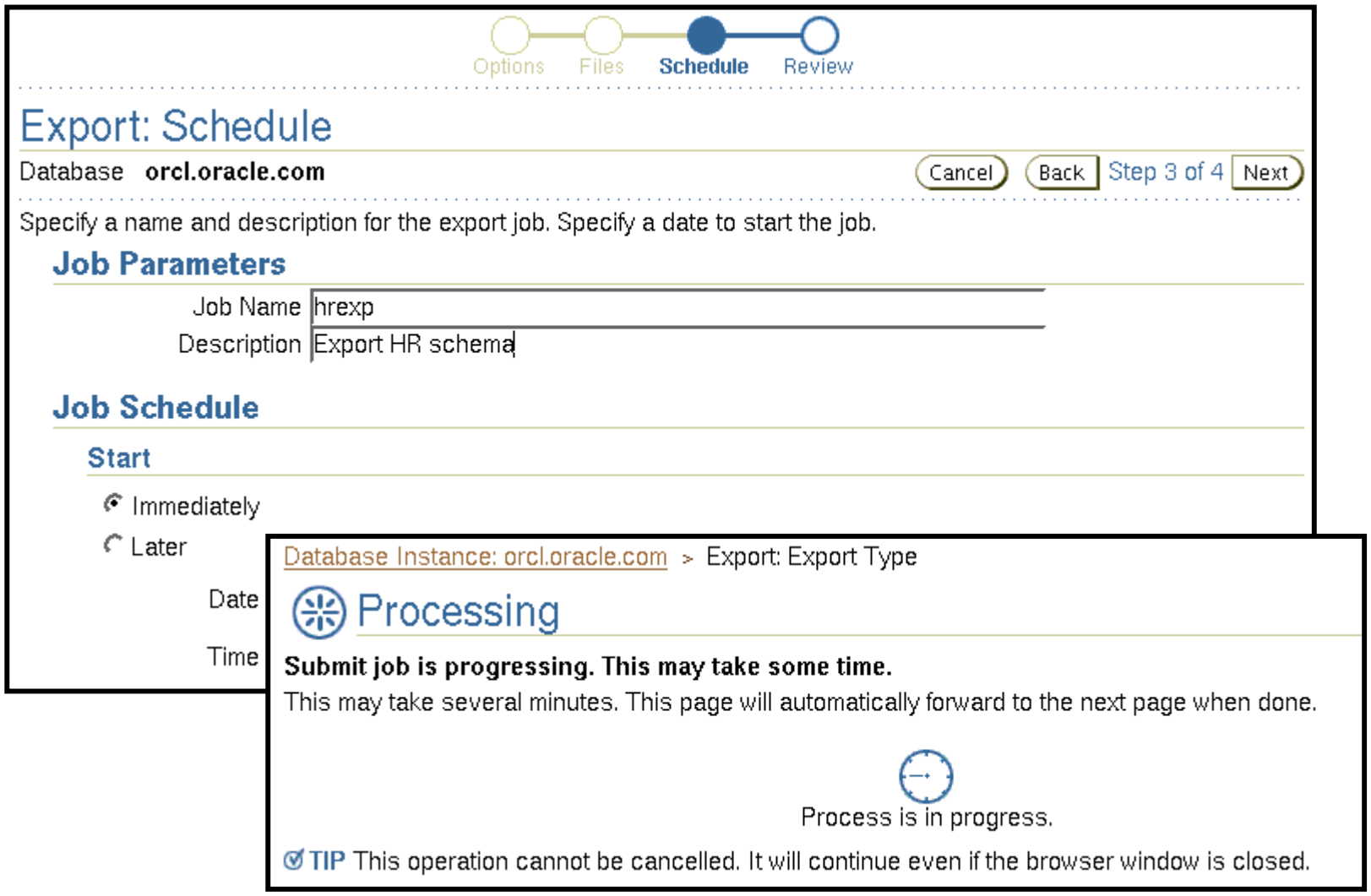
# Data Pump File Locations



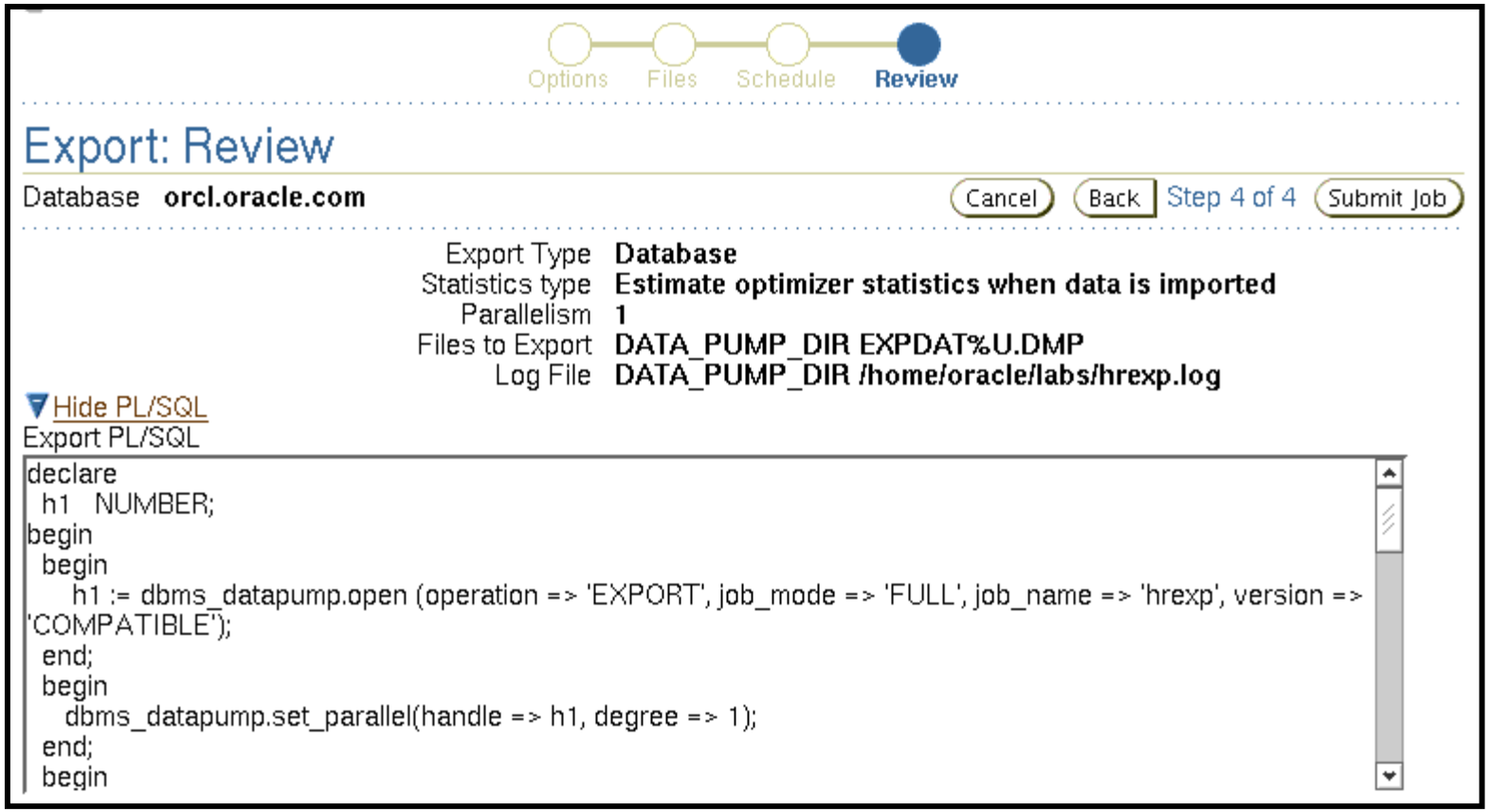

**The order of precedence of file locations:**

- **Per-file directory**
- **The `DIRECTORY` parameter**
- **The `DATA_PUMP_DIR` environment variable**
- **`DATA_PUMP_DIR` directory object**

# Scheduling and Running a Job



# Data Pump File Naming and Size



# Data Pump Import

- Directory Obj.
- SQL*Loader
- Data Pump
  - Export
> - Import
- External Table

# Data Pump Import: Transformations

You can remap:

- Data files by using `REMAP_DATAFILE`
- Tablespaces by using `REMAP_TABLESPACE`
- Schemas by using `REMAP_SCHEMA`

```
REMAP_DATAFILE = 'C:\oradata\tbs6.f':'/u1/tbs6.f'
```

# Data Pump Import: Transformations

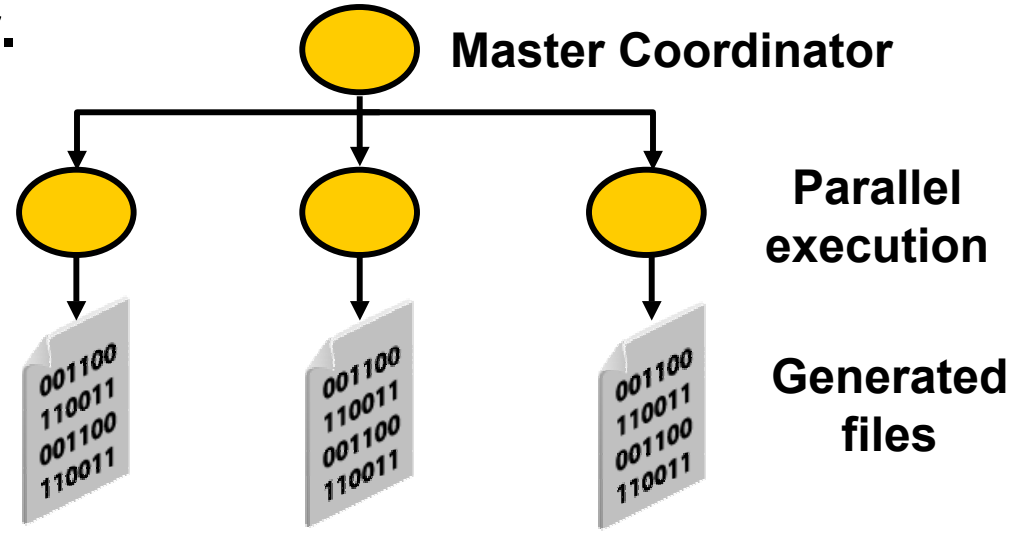
Using `TRANSFORM`, you can also :

- Exclude from tables and indexes:
  - `STORAGE` and `TABLESPACE` clauses
  - `STORAGE` clause only
- Re-create object identifiers of abstract data types
- Change extent allocations and file size

```
TRANSFORM =
SEGMENT_ATTRIBUTES|STORAGE|OID|PCTSPACE:{y|n|v}[:object type]
```

# Data Pump: Performance Consideration

**Maximizing job performance with the `PARALLEL` parameter.**




**Example:**

```
expdp hr/hr FULL=y
DUMPFILE=dp_dir1:full1%U.dmp, dp_dir2:full2%U.dmp
FILESIZE=2G PARALLEL=3
LOGFILE=dp_dir1:expfull.log JOB_NAME=expfull
```

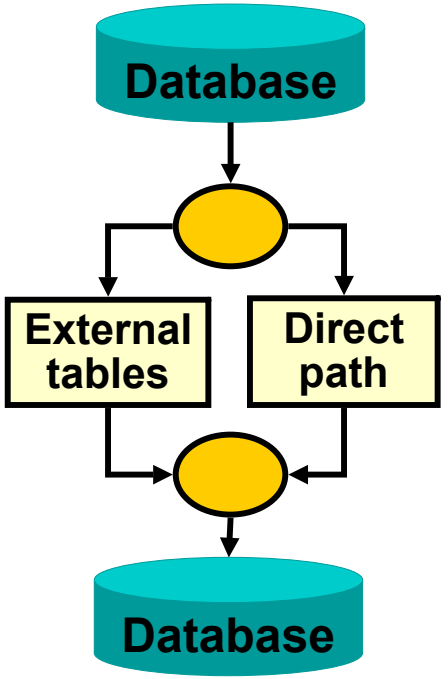
# Performance Initialization Parameters

- **Performance of Data Pump can be affected by:**
  - `DISK_ASYNCH_IO=TRUE`
  - `DB_BLOCK_CHECKING=FALSE`
  - `DB_BLOCK_CHECKSUM=FALSE`
- **The following should be set high to allow for maximum parallelism:**
  - `PROCESSES`
  - `SESSIONS`
  - `PARALLEL_MAX_SERVERS`
- **The following should be sized generously:**
  - `SHARED_POOL_SIZE`
  - `UNDO_TABLESPACE`

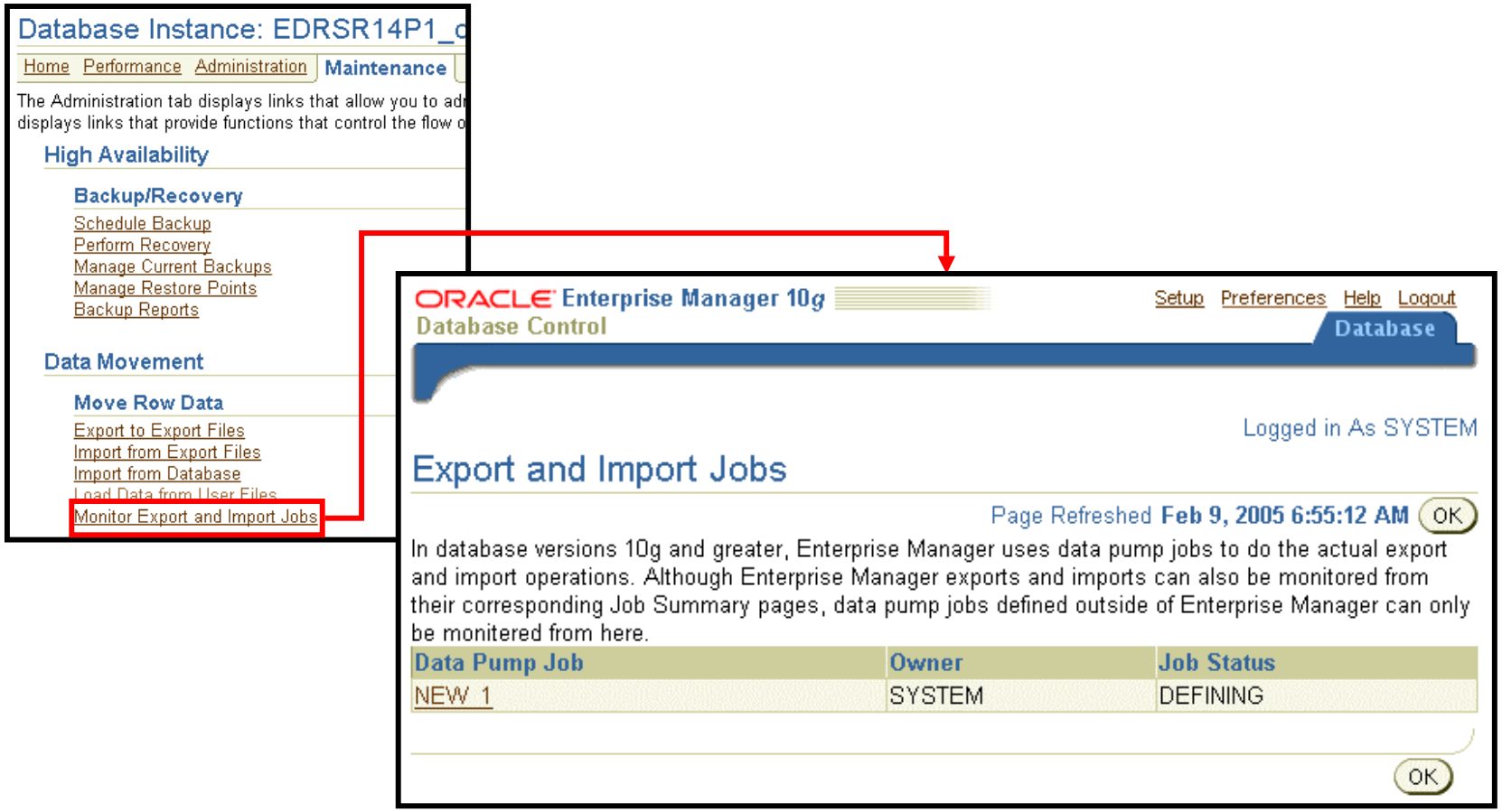
# Data Pump Access Path: Considerations

One of the following access paths is automatically selected by Data Pump:

- Direct path
- External tables, if data includes:
  - Encrypted columns
  - Clustered tables
  - Different partition at unload and load time, and others (see Notes)



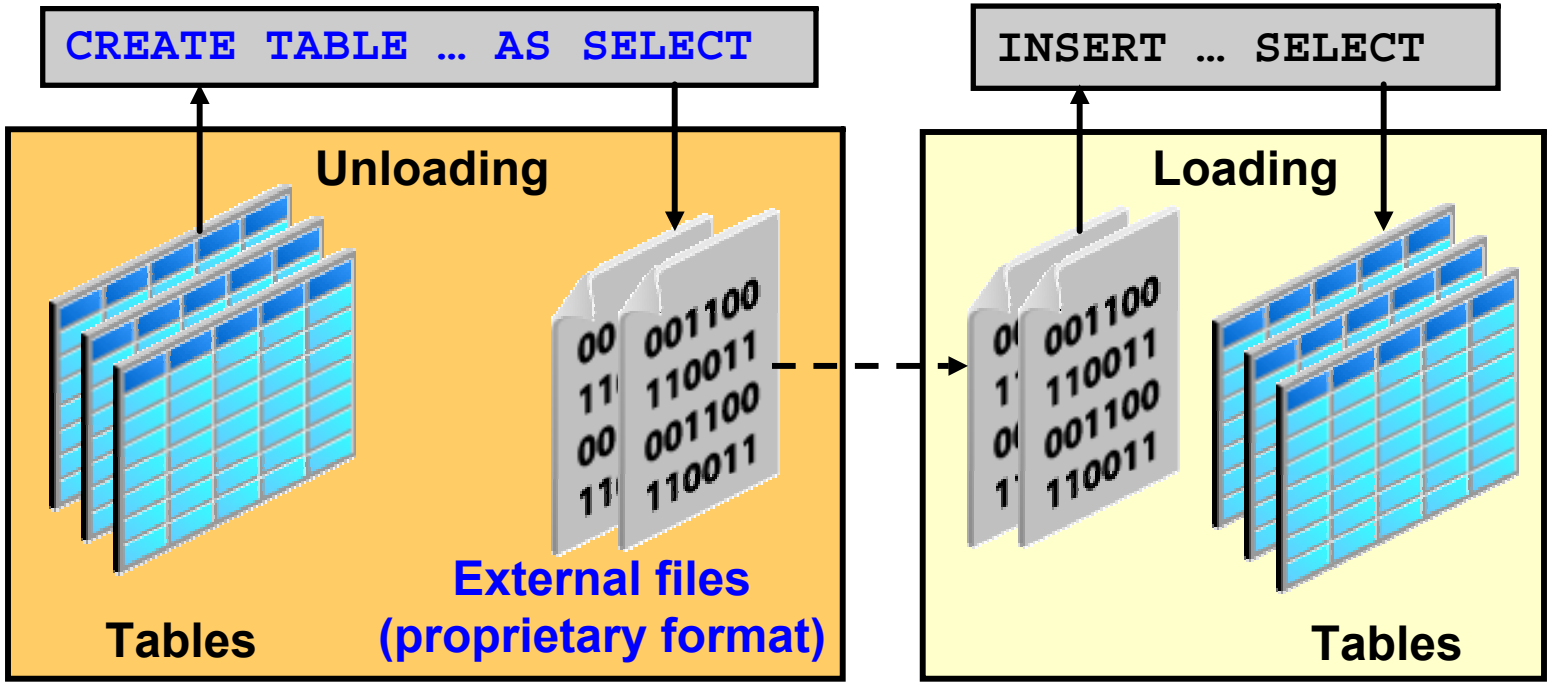
# Using Enterprise Manager to Monitor Data Pump Jobs



# External Table Population

- Directory Obj.
- SQL*Loader
- Data Pump
  - Export
  - Import
- > External Table

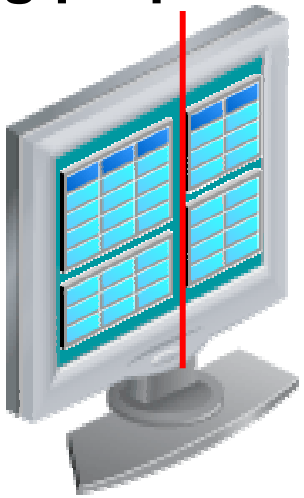
- Unloading of data to external files with the `ORACLE_DATAPUMP` access driver
- No modifications of external tables



# Using External Tables

- Data can be used directly from the external file or loaded into another database.
- Resulting files can be read only with the `ORACLE_DATAPUMP` access driver.
- You can combine generated files from different sources for loading purposes.

From Oracle Database



From External File

# External Table Population with ORACLE_DATAPUMP

```
CREATE TABLE emp_ext
  (first_name, last_name, department_name)
ORGANIZATION EXTERNAL
  (
    TYPE ORACLE_DATAPUMP
    DEFAULT DIRECTORY ext_dir
    LOCATION ('emp1.exp','emp2.exp','emp3.exp')
  )
PARALLEL
AS
SELECT e.first_name,e.last_name,d.department_name
FROM   employees e, departments d
WHERE  e.department_id = d.department_id AND
       d.department_name in
                    ('Marketing', 'Purchasing');
```

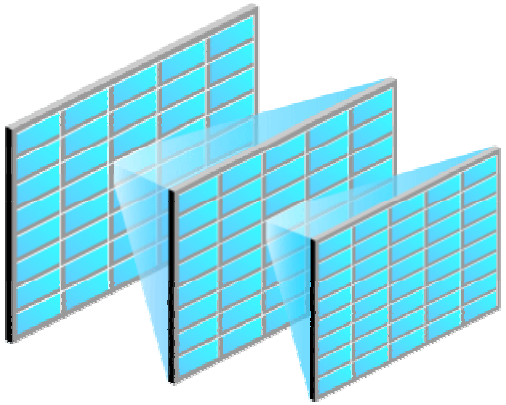
# External Table Population with ORACLE_LOADER

```
CREATE TABLE extab_employees
                    (employee_id        NUMBER(4),
                     first_name         VARCHAR2(20),
                     last_name          VARCHAR2(25),
                     hire_date          DATE)
ORGANIZATION EXTERNAL
     ( TYPE ORACLE_LOADER DEFAULT DIRECTORY extab_dat_dir
       ACCESS PARAMETERS
       ( records delimited by newline
         badfile extab_bad_dir:'empxt%a_%p.bad'
         logfile extab_log_dir:'empxt%a_%p.log'
         fields terminated by ','
         missing field values are null
     ( employee_id, first_name, last_name,
      hire_date char date_format date mask "dd-mon-yyyy"))
     LOCATION ('empxt1.dat', 'empxt2.dat') )
     PARALLEL  REJECT LIMIT UNLIMITED;
```

# Data Dictionary

**View information about external tables in:**

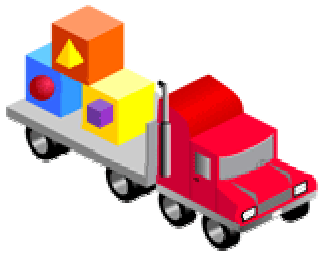
- `[DBA | ALL | USER]_EXTERNAL_TABLES`
- `[DBA | ALL | USER]_EXTERNAL_LOCATIONS`
- `[DBA | ALL | USER]_TABLES`, **and others**



# Summary

In this lesson, you should have learned how to:

- Describe available ways for moving data
- Create and use directory objects
- Use SQL*Loader to load data from a non-Oracle database (or user files)
- Explain the general architecture of Data Pump
- Use Data Pump Export and Import to move data between Oracle databases
- Use external tables to move data via platform-independent files



# Practice Overview: Moving Data

This practice covers the following topics:

- Using the Data Pump Export Wizard to select database objects to be exported
- Monitoring a Data Pump Export job
- Using the Data Pump Import Wizard to import tables in your database
- Using the Load Data Wizard to load data into your database
- Loading data by using the command line