

ORACLE®

Data Pump, sql loader, TTS и многое другое

Содержание

- Почему мы заменяем exp и imp?
- Что такое Data Pump?
- Использование expdp и impdp
- SQL Loader
- Внешние таблицы



Прочему мы заменяем ехр и ітр?





Прочему мы заменяем ехр и ітр?

Ограничения ехр, ітр

- Медленная загрузка, выгрузка
- ехр и ітр работают в одном процессе
- Нельзя рестартовать
- ітр читает файлы последовательно
- ехр и ітр имеют ограниченный набор правил для отбора объектов
- ітр имеет ограниченный набор трансформаций
- Не поддержка для сжатия и шифрования
- Нужен “безфайловый” режим

Что такое Data Pump?





Что такое Data Pump?

Определение

- Инфраструктура в базе данных, используемая для выгрузки и загрузки метаданных и данных
- Клиенты - expdp и impdp



Что такое Data Pump?

Компоненты

- Утилиты expdp и impdp
- PL/SQL Package DBMS_DATAPUMP
- PL/SQL Package DBMS_METADATA
- Внешние таблицы с доступом через SQL
- Direct path механизм загрузки, с доступом через OCI



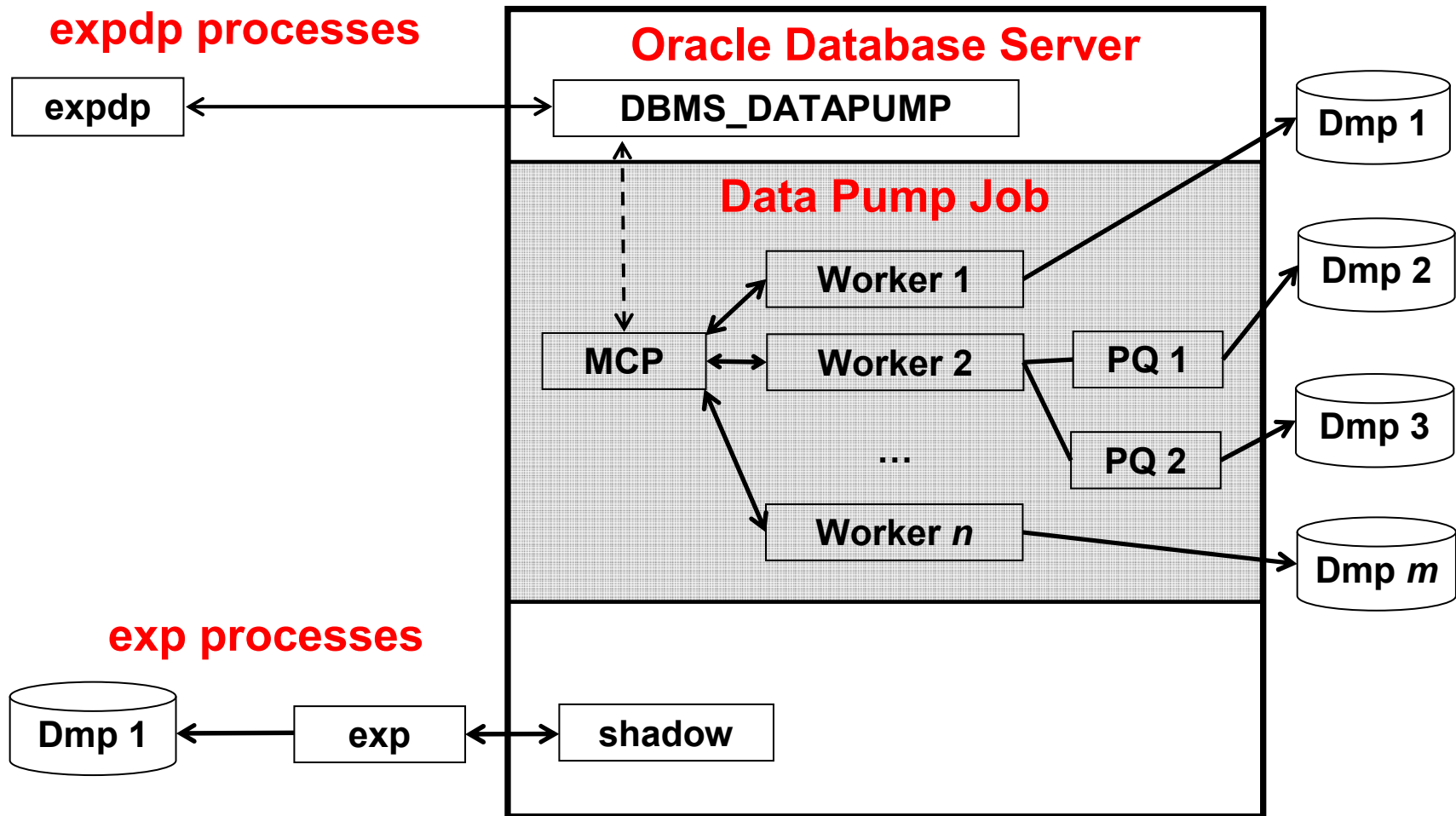
Что такое Data Pump?

Процессы

- На клиенте работают `expdp` и `impdp`
- Серверный процесс взаимодействует с клиентом и запускает фоновые процесс
- Master Control Process (MCP)
- Рабочие процессы – много, если `parallel`
- Параллельные серверные процессы (if used)
- На RAC, MCP и Worker все на одном узле; параллельные процессы могут быть на других узлах

Что такое Data Pump?

Сравнение exp и expdp



Использование exrdr and imrdr





Использование expdp and impdp

Клиент/серверная архитектура

- Операции Data Pump запускаются на сервере
- Dump, log и SQL файлы находятся на сервере (в DIRECTORY)
- Клиент лишь отображает сообщения сервера
- Клиент может посылать команды серверу
- Операция продолжает выполняться после отключения клиента



Using expdp and impdp

Export and Import Modes – New in Data Pump

- Network Mode
 - Network export unloads data and metadata from a remote database into a dump file over a dblink
 - Network import copies data and metadata from a remote database into local database over a dblink without using a dump file
- Table Transportable (11.1)
 - Table export/import using transportable tablespaces
 - Export writes metadata to dump file and lists tablespaces that need to be copied
 - Import recreates metadata and plugs tablespace into new database. Space not used by table or dependent objects is freed



Using expdp and impdp

Naming Files – Directory Objects

- System manager creates directory objects in SQL
 - Directory object containing dump files for HR:
CREATE DIRECTORY HR_DMP AS '/hr/files/dmp';
 - Directory object containing log files for HR:
CREATE DIRECTORY HR_LOG AS '/hr/files/log';
- Use GRANT to allow access to other users
 - Let HR read, but not write dump files; restricts HR to impdp:
GRANT READ ON DIRECTORY HR_DMP TO HR;
 - Let HR read and write log files:
GRANT READ,WRITE ON DIRECTORY HR_LOG TO HR;



Using expdp and impdp

Naming Files - examples

- Old exp command
 - `exp sys/*** file=/hr/files/dump/hr.dmp owner=hr log=/hr/files/log/hr_exp.log`
- New expdp command
 - `expdp sys/*** dumpfile=hr_dmp:hr.dmp schema=hr logfile=hr_log:hr_exp.log`
- Old imp command
 - `imp hr/*** dumpfile=/hr/files/dump/hr.dmp tables=employees log=/hr/files/log/emp_imp.log`
- New impdp command
 - `impdp hr/*** dumpfile=hr_dmp:hr.dmp tables=employees logfile=hr_log:emp_imp.log`



Using expdp and impdp

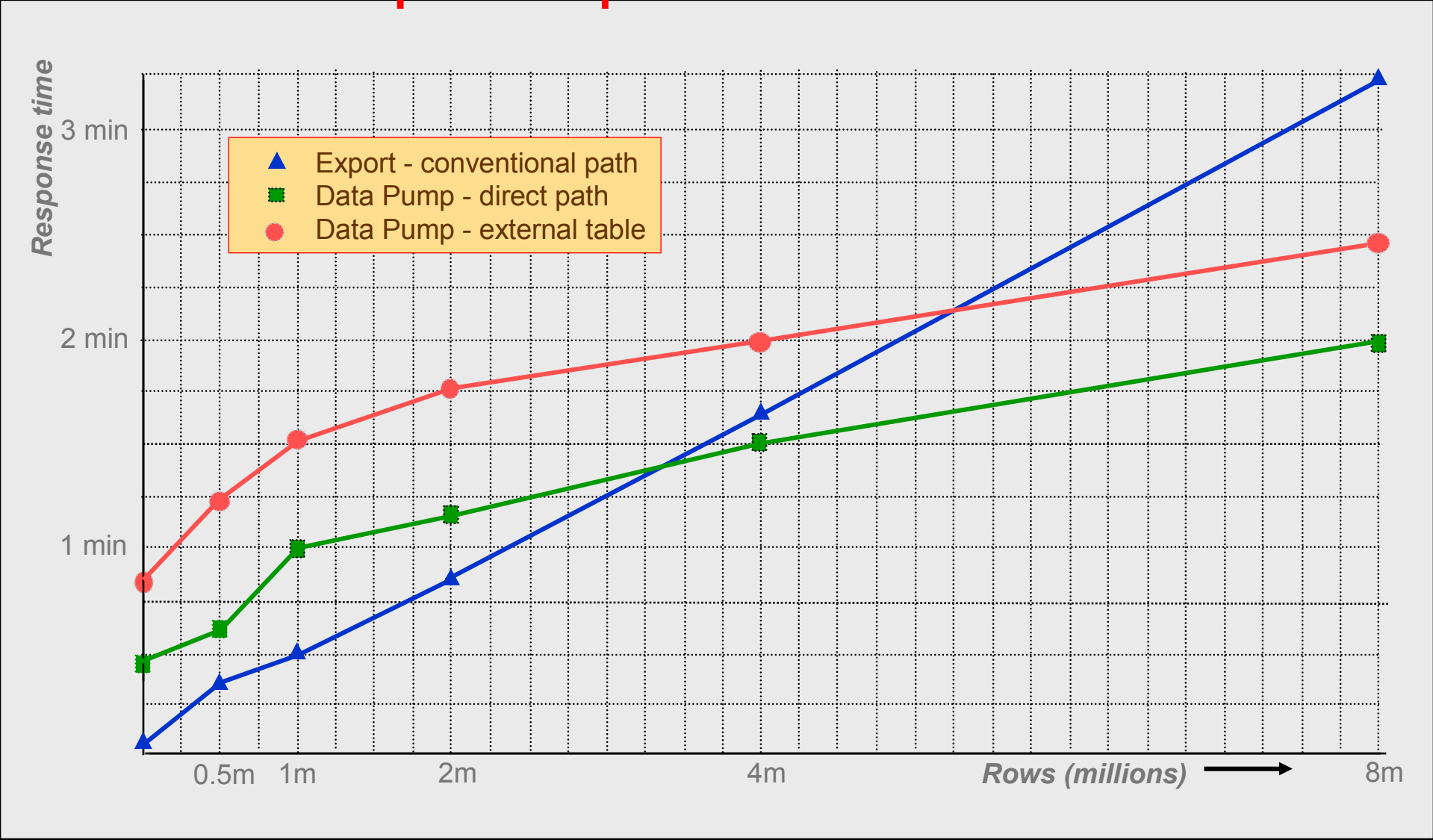
Specifying Objects – CONTENT parameter

- CONTENT (DATA_ONLY, METADATA_ONLY, ALL)
- INCLUDE, EXCLUDE
 - An object type; INDEX
 - An object type and an object name; INDEX:"EMPID"
 - An object type and a SQL expression for the object name; INDEX:"LIKE EMP%"
- REMAP_DATAFILE, REMAP_SCHEMA, REMAP_TABLESPACE
- TRANSFORM: SEGMENT_ATTRIBUTES, STORAGE и т.д.
- ADD_FILE, PARALLEL, START_JOB, STATUS



Going Forward

What to expect – exp Performance

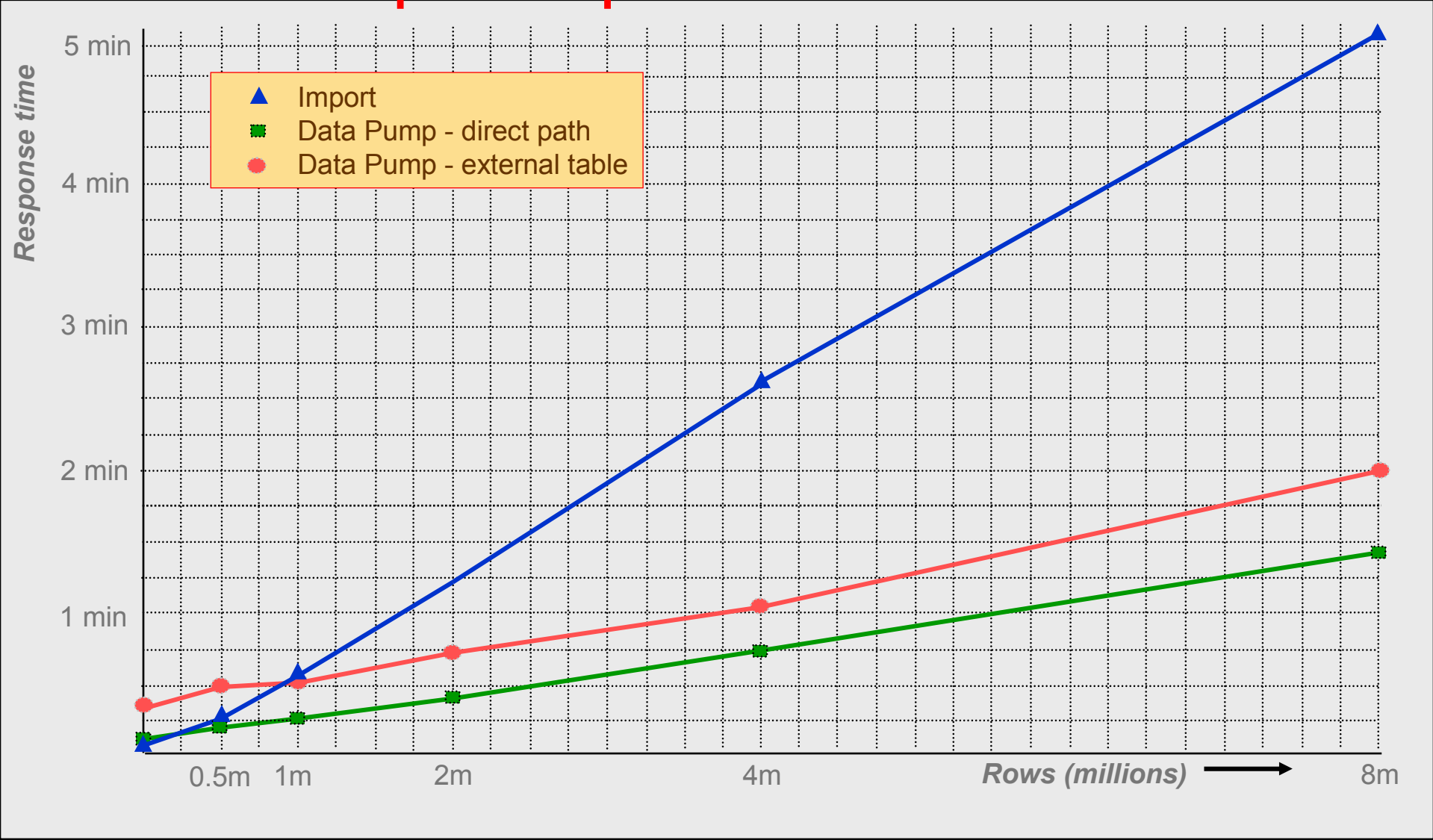


Test Results Courtesy of Prof. Carl Dudley, University of Wolverhampton, UK



Going Forward

What to expect – imp Performance

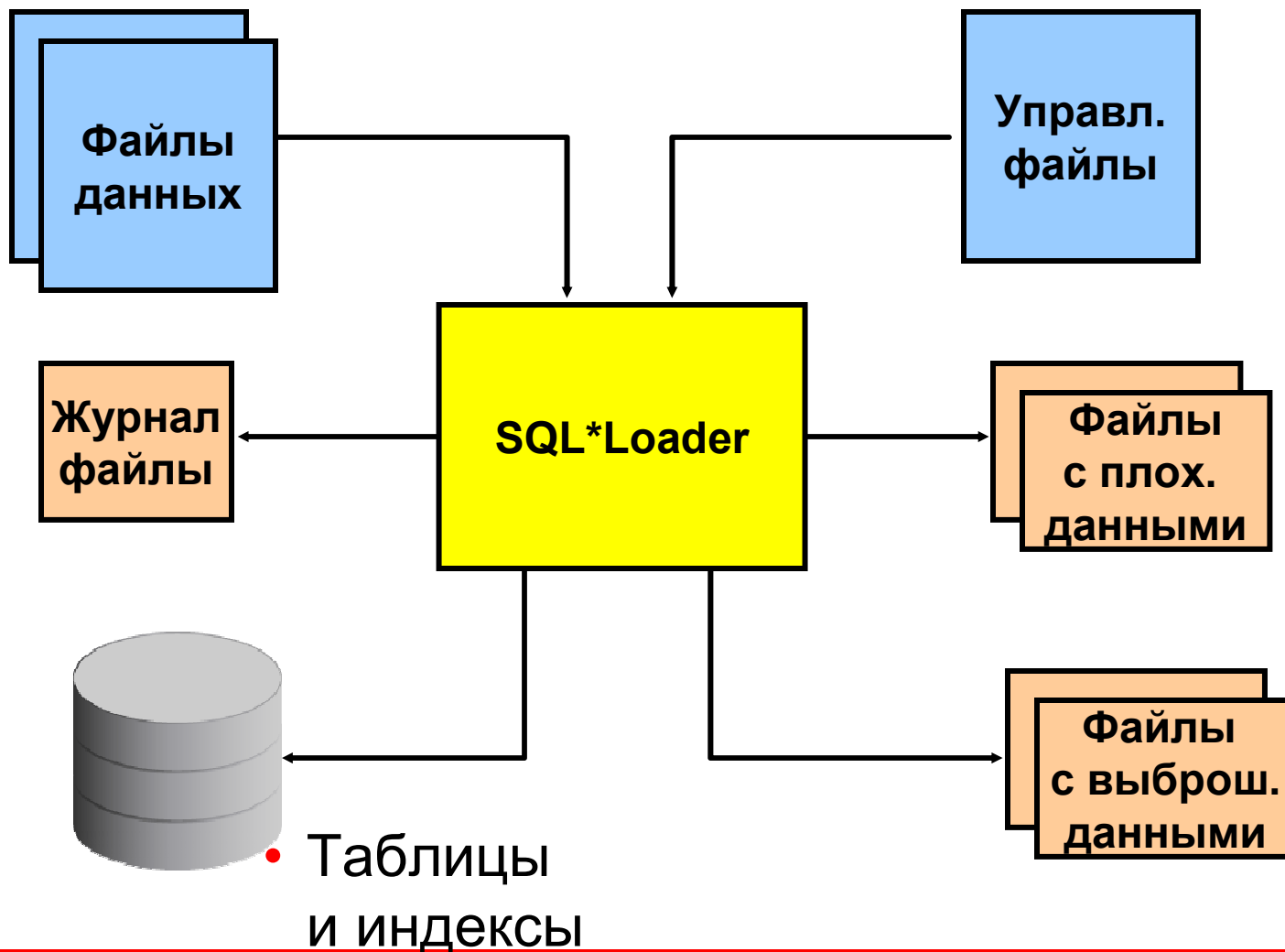


Test Results Courtesy of Prof. Carl Dudley, University of Wolverhampton, UK

SQL Loader



SQL*Loader



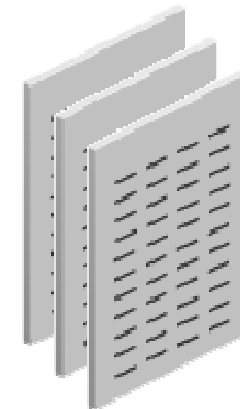
■ Управляющий файл SQL*Loader

- Содержит информацию:
 - Где искать данные
 - Формат данных
 - Детали конфигурации:
 - Управление памятью
 - Работу с некорректными записями
 - Interrupted load handling details
 - И т.д.

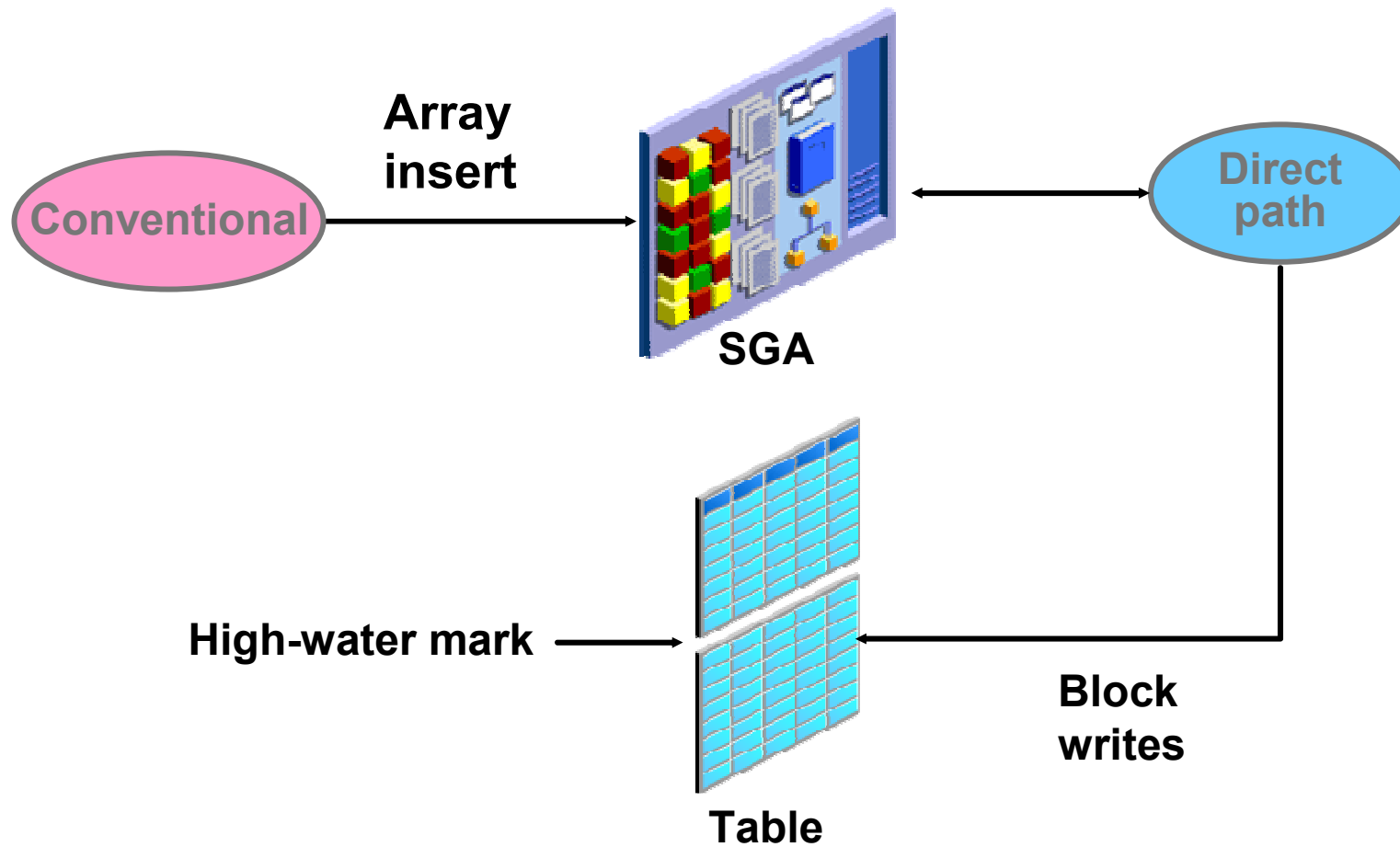


Файлы данных

- SQL*Loader читает данные из одного или нескольких файлов данных.
- С точки зрения SQL*Loader's данные организованы как записи.
- Файл данных может быть в 3х форматах:
 - Фиксированная длина записи
 - Переменная длина записи
 - Поточковые



Методы загрузки





Direct и Conventional способы загрузки

Conventional Load	Direct Path Load
Uses COMMIT to make changes permanent	Uses data saves
Redo entries always generated	Generates redo only under specific conditions
Enforces all constraints	Enforces only PRIMARY KEY, UNIQUE, and NOT NULL
INSERT triggers fire	INSERT triggers do not fire
Can load into clustered tables	Cannot load into clustered tables
Other users can make changes to tables	Other users cannot make changes to tables

Пример использования SQL*Loader

- Пример данных SALES:

```
salesDec01.dat - Notepad
File Edit Format View Help
133|6538|05-DEC-2001|P|999|1|33.25|
133|702|06-DEC-2001|P|999|1|33.25|
133|2671|06-DEC-2001|P|999|1|33.25|
133|6105|07-DEC-2001|I|999|1|32.27|
```

- Управляющий файл для таблицы SALES:

```
• LOAD DATA INFILE salesDec01.dat APPEND INTO TABLE sales
• FIELDS TERMINATED BY "|"
• (PROD_ID, CUST_ID, TIME_ID, CHANNEL_ID, PROMO_ID,
  QUANTITY_SOLD, AMOUNT_SOLD)
```

- Таблица может быть загружена:

```
• $ sqlldr sh/sh control=sales_dec01.ctl direct=true
```


EM поддерживает SQL*Loader

ORACLE Enterprise Manager 10g Database Control

Setup Preferences Help Logout

Database

Control File Data File Load Method Options Schedule Review

Load Data: Control File

Database **orcl.oracle.com** Cancel Finish Step 1 of 6 Next

A control file is used to describe what will be loaded and how. Specify the full path and name of the control file on the database server machine.

Host Credentials

* Username

* Password

Save as Preferred Credential

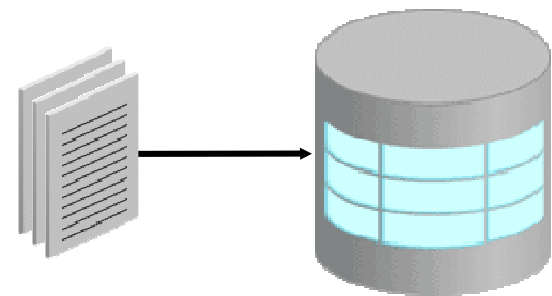


Внешние таблицы



Использование внешних таблиц

- Внешние таблицы это таблицы, которые хранят данные во внешних файлах (текст или бинарный)
 - Доступ такой же как к обычным таблицам
 - DML и создание индексов запрещено
 - Метаданные создаются командой `CREATE TABLE`
 - Внешняя таблица описывает, как данные должны быть представлены в базе.





Плюсы внешних таблиц

- Запросы и соединения без загрузки в базу
- Полезно, когда внешние данные нужно соединить с данными в базе и затем трансформировать
- Полезно, когда внешние данные запрашиваются нечасто
- Параллелизм не требует разбивать входной файл на несколько
- Дополняет SQL*Loader функционал:
 - Прозрачные параллелизм
 - Все возможности SQL для direct-path загрузки

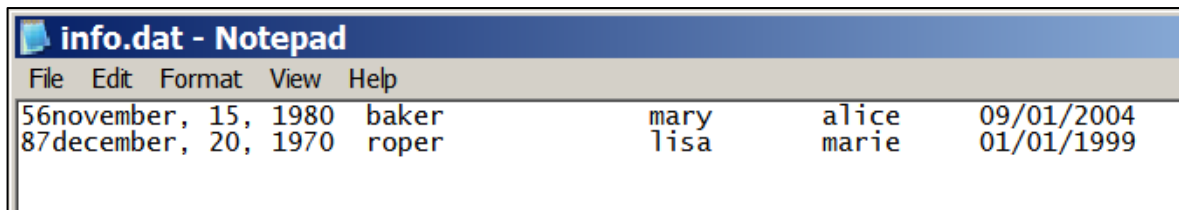


Создание внешней таблицы

- `CREATE TABLE...ORGANIZATION EXTERNAL :`
 - `TYPE: ORACLE_LOADER` или `ORACLE_DATAPUMP` драйвер
 - `DEFAULT DIRECTORY:` указывает место, где по-умолчанию лежат файлы
 - `ACCESS PARAMETERS:` Параметры, описывающие внешний файл
 - `LOCATION:` Местоположение данных

Создание и загрузка внешних таблиц: ORACLE_LOADER (1)

1. Определить структуру файла:



The screenshot shows a Notepad window titled 'info.dat - Notepad'. The window contains the following text:

File	Edit	Format	View	Help	
56	november, 15, 1980	baker	mary	alice	09/01/2004
87	december, 20, 1970	roper	lisa	marie	01/01/1999

2. Настроить директории и доступ к ним:

```
CREATE DIRECTORY ext_tab_dir AS '/usr/apps/datafiles';  
GRANT READ ON DIRECTORY ext_tab_dir TO sh;
```

Создание и загрузка внешних таблиц: ORACLE_LOADER (2)

3. Создание таблицы emp_load:

```
CREATE TABLE emp_load
  (employee_number    CHAR(5), employee_dob    CHAR(20),
   employee_last_name CHAR(20), employee_first_name
   CHAR(15),
   employee_middle_name CHAR(15), employee_hire_date    DATE)
ORGANIZATION EXTERNAL          -- External table
(TYPE ORACLE_LOADER          -- Access driver
DEFAULT DIRECTORY def_dir1   -- Files directory
ACCESS PARAMETERS            -- Similar to SQL*Loader
 (RECORDS DELIMITED BY NEWLINE
  FIELDS (employee_number    CHAR(2),
         employee_dob        CHAR(20),
         employee_last_name  CHAR(18),
         employee_first_name CHAR(11),
         employee_middle_name CHAR(11),
         employee_hire_date  CHAR(10) date_format DATE mask
         "mm/dd/yyyy") ) LOCATION ('info.dat'));
```

Создание и загрузка внешних таблиц: ORACLE_LOADER (3)

4. Загрузка данных из внешней таблицы emp_load в новую таблицу emp:

```
CREATE TABLE emp (emp_no CHAR(6), last_name CHAR(25),  
first_name CHAR(20), middle_initial CHAR(1),  
hire_date DATE, dob DATE) AS SELECT employee_number,  
employee_first_name,  
substr(employee_middle_name, 1, 1), employee_last_name,  
employee_hire_date, to_date(employee_dob, 'month, dd,  
yyyy')  
FROM emp_load;
```

```
SELECT * FROM emp;
```

EMP_NO	LAST_NAME	FIRST_NAME	M	HIRE_DATE	DOB
56	baker	mary	a	01-SEP-04	15-NOV-80
87	roper	lisa	m	01-JAN-99	20-DEC-70

```
2 rows selected.
```



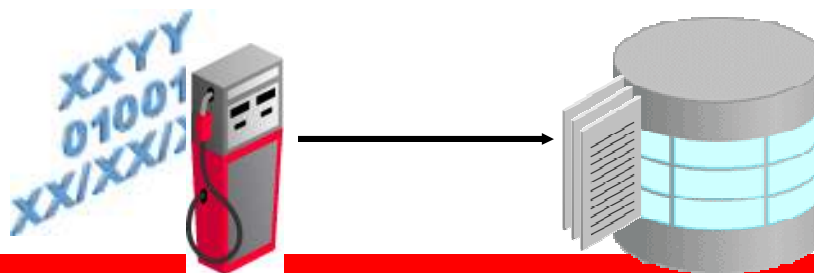

Генерация команды для внешних таблиц в SQL*Loader

- Если уже есть управляющий файл SQL*Loader, то по нему очень легко можно получить скрипт для создания внешних таблиц:

```
$ sqlldr sh/sh control=sales_dec00.ctl  
EXTERNAL_TABLE=GENERATE_ONLY LOG=sales_dec00.sql
```

Создание и загрузка внешних таблиц: ORACLE_DATAPUMP

```
CREATE TABLE emp_ext
  (first_name, last_name, department_name)
ORGANIZATION EXTERNAL
  ( TYPE ORACLE_DATAPUMP
    DEFAULT DIRECTORY ext_dir
    LOCATION ('emp1.exp',) )
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');
```





Изменение свойств внешней таблицы

- Используя ALTER TABLE можно менять:
 - DEFAULT DIRECTORY
 - ACCESS PARAMETERS
 - LOCATION
 - REJECT_LIMIT
 - Support for parallel processing

```
ALTER TABLE sales_delta_xt LOCATION ('newfile1.dat')
```

Transportable Table Spaces





Transportable Tablespaces

- Это самый быстрый способ переноса больших объемов данных.
- Исходная и целевая базы могут иметь разный размер блока
- Метод отлично подходит для переноса данных с OLTP в хранилище данных.
- До Oracle Database 10g, исходная и целевая базы должны были использовать одинаковые операционные системы.

Transportable Tablespaces: Пример

1. Помещаем данные в tablespace:

```
CREATE TABLE temp_jan_sales NOLOGGING
TABLESPACE ts_temp_sales AS SELECT * FROM sales
WHERE time_id BETWEEN '31-DEC-1999' AND '01-FEB-2000';
```

2. Проверяем самодостаточность tablespace:

```
DBMS_TTS.TRANSPORT_SET_CHECK(TS_LIST=>'ts_temp_sal' ,
INCL_CONSTRAINTS=>TRUE);
SELECT * FROM TRANSPORT_SET_VIOLATIONS;
```

3. В режим «только чтение»

```
ALTER TABLESPACE ts_temp READ ONLY;
```

4. Экспортируем, копируем и импортируем данные:

```
EXPDP DIRECTORY=DW_DUMP_DIR DUMPFILE=jan.dmp
TRANSPORT_TABLESPACES=ts_temp_sales
CP ...
IMPDP DIRECTORY=DM_DUMP_DIR DUMPFILE=jan.dmp
TRANSPORT_DATAFILES='/db/tempjan.f'
```

Transportable Tablespaces: Пример

5. Добавляем данные из `temp_sales_jan` в таблицу:

```
-- Option 1: Append data directly into destination table.
```

```
INSERT /*+ APPEND */ INTO sales  
SELECT * FROM temp_sales_jan;
```

```
-- Option 2: Alternatively, if the data mart's sales table is  
-- partitioned accordingly then the temp_sales_jan table can  
-- be exchanged with a new partition using the partition  
-- exchange feature.
```

```
ALTER TABLE sales ADD PARTITION sales_00_jan VALUES  
LESS THAN (TO_DATE('01-feb-2000', 'dd-mon-yyyy'));  
ALTER TABLE sales EXCHANGE PARTITION sales_00_jan  
WITH TABLE temp_sales_jan INCLUDING INDEXES WITH  
VALIDATION;
```



Resumable Statements

- A resumable statement:
 - Позволяют «усыпить» длинную операцию вместо получения ошибки
 - Дают шанс исправить решить проблему пока операция «спит», а не начинать операцию заново
 - Операция может уснуть, если :
 - Не хватает места
 - Достигнуто максимальное количество экстенентов
 - Превышена квота

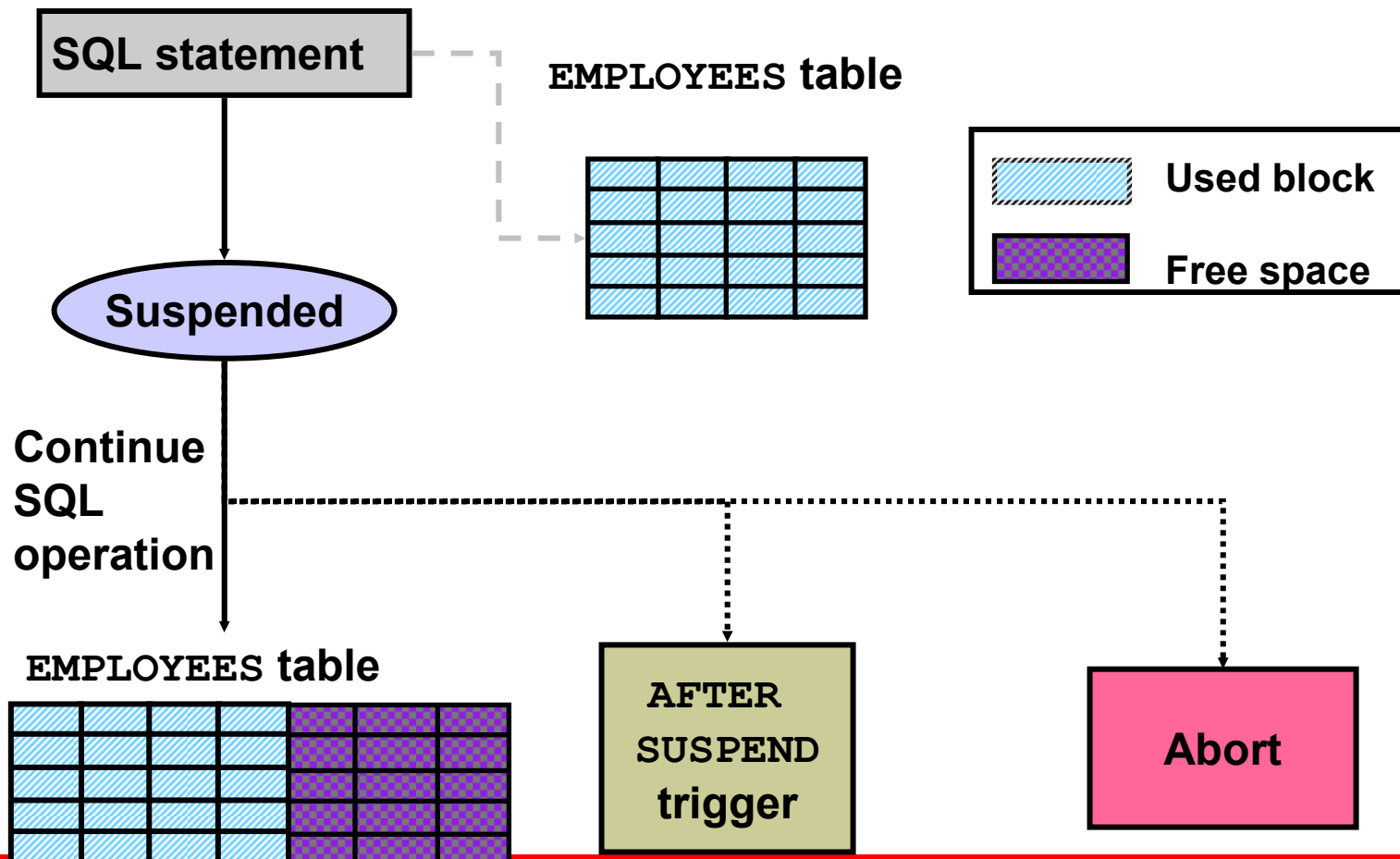


Использование Resumable Space Allocation

- Запросы, DML операции, и некоторые DDL операции могут быть возобновлены после ошибки, связанной с нехваткой места.
- Сессия может быть открыта через различные интерфейсы SQL, PL/SQL, SQL*Loader, OCI
- Для запуска возобновляемых операций необходимо сделать настройки

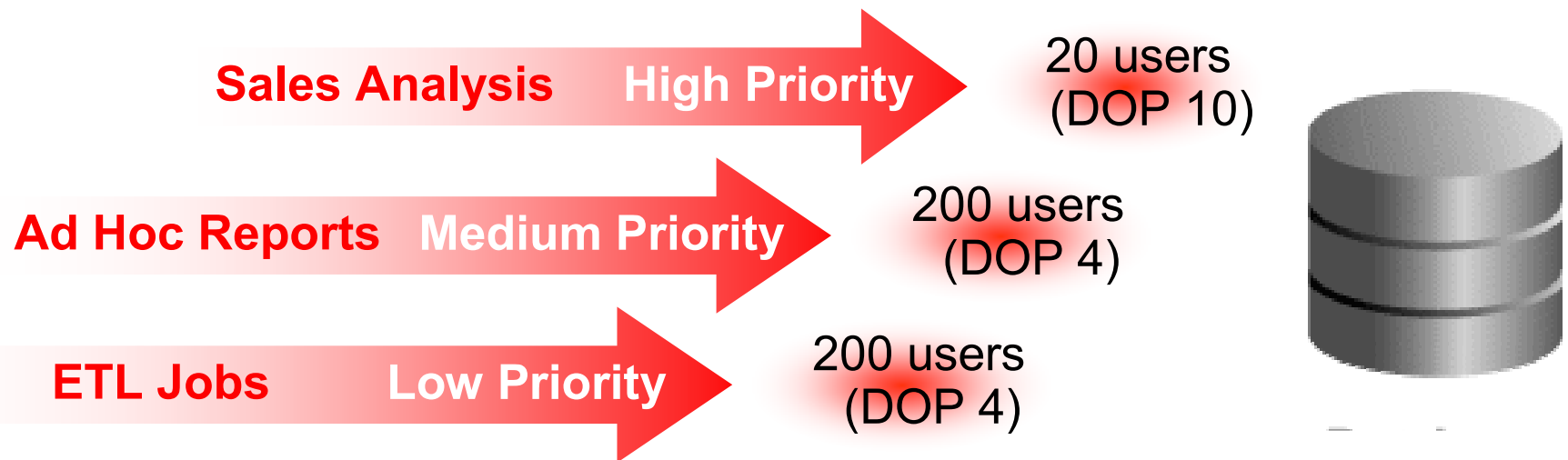
```
ALTER SESSION ENABLE RESUMABLE ;  
ALTER SESSION SET RESUMABLE_TIMEOUT=3600 ;  
  
INSERT INTO sales_new SELECT * FROM sh.sales ;  
  
ALTER SESSION DISABLE RESUMABLE ;
```

Возобновление «усыпленной» операции



Database Resource Manager

- Protect the system pro-actively
 - Maximum number of concurrent operations
 - Priority-dependent maximum Degree Of Parallelism (DOP)



SQL Query Result Cache

- Store query results in cache
 - Repetitive executions can use cached result
- Data Warehouse queries
 - Long-running, IO-intensive
 - Expensive computations
 - Return few rows
 - Excellent opportunity for SQL Query Result Cache

Id	Operation	Name
0	SELECT STATEMENT	
1	RESULT CACHE	fz6cm4jbpchw48wcyk60m7qypu
2	SORT GROUP BY ROLLUP	
* 3	HASH JOIN	

etc.

SQL Query Result Cache Opportunity

- Retail customer data (~50 GB)
 - Concurrent users submitting queries randomly
 - Executive dashboard with 12 heavy analytical queries
 - Cache results only at in-line view level
 - 12 queries run in random, different order – 4 queries cached
- Measure average, total response time for all users

# Users	No cache	Cache	Improvement
2	186 s	141 s	24%
4	267 s	201 s	25%
8	447 s	334 s	25%