

24

在数据库之间传输数据

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

目标

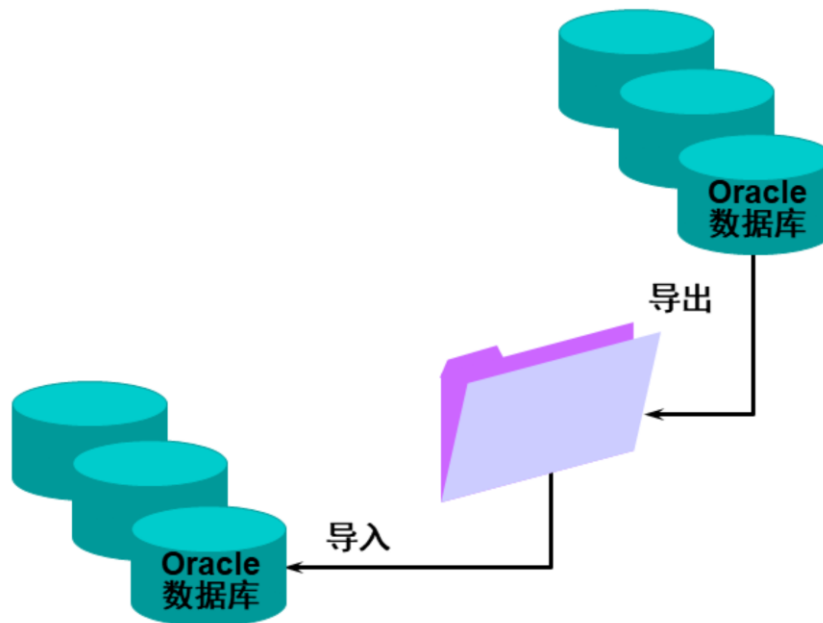
完成这一课的学习后，您应该能达到下列目标：

- 描述导出和导入实用程序的使用
- 描述导出和导入的概念与结构
- 执行简单的导出和导入操作
- 列出使用导出和导入操作的原则

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

Oracle 导出和导入实用程序



ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

导出和导入实用程序

导出实用程序提供了一种简单方法，使用该方法，您可以在 Oracle 数据库之间传送数据对象，即使这些数据库位于具有不同硬件和软件配置的平台之上。

对 Oracle 数据库运行导出实用程序时，该实用程序将提取对象（如表），如果这些对象还存在相关对象，则继续提取这些相关对象（如索引、注释和授权）。提取出的数据写入一个导出文件中，该文件是一个 Oracle 二进制格式的转储文件，且通常位于磁盘或磁带上。

导入实用程序从导出转储文件中读取对象定义和表数据。它用于将数据对象插入 Oracle 数据库。

Oracle 导出和导入实用程序概览

您可以使用这两个实用程序来执行以下任务：

- 归档历史数据
- 保存表定义以防止出现用户错误故障
- 在计算机和数据库之间，或在不同版本的 Oracle 服务器之间移动数据
- 在数据库之间传输表空间

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

导出和导入实用程序概览

导出实用程序可以提供以下内容的逻辑备份：

- 数据库对象
- 表空间
- 整个数据库

导入实用程序用于读取有效的导出文件，以便将数据移动到数据库中。重做日志历史记录不能应用于从导出文件中导入的对象，因此，虽然仍可能会造成数据丢失，但是可以使丢失减至最少。

您可以使用导出和导入实用程序来执行正常的操作系统备份，方法是使用这两个实用程序执行以下操作：

- 创建某个数据库对象或整个数据库的历史归档；例如，在修改某个方案以支持不断变化的业务要求时。
- 将表定义保存在一个二进制文件中。这对于创建和维护一个给定方案结构的基线可能非常有用。
- 将数据从一个 Oracle 数据库版本移动到另一个 Oracle 数据库版本，例如升级到 Oracle11g。

导出和导入实用程序概览（续）

使用这两个实用程序可以防止出现以下情形：

- 可能因用户意外删除或截断表而造成的用户错误
- 逻辑上已经损坏的表
- 只影响数据库的子集的不正确的批处理作业或其它 DML 语句

使用这两个实用程序可以恢复：

- 当一个物理数据库的各个表空间中存在多个逻辑数据库时，将某个逻辑数据库恢复到与物理数据库中其它逻辑数据库不同的时间点
- 当表空间时间点恢复 (TSPITR) 与使用备份还原整个数据库然后再前滚数据库相比，前者的效率更高时，则恢复超大型数据库 (VLDB) 中的某个表空间

注：本课程介绍导出和导入实用程序，并讨论这两个实用程序如何影响备份和恢复操作。有关这两个实用程序的详细说明，请参阅 *Oracle Database Utilities* 手册。

调用导出和导入实用程序的方法

- 命令行界面
- 交互式对话
- 参数文件
- **Oracle Enterprise Manager**

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

调用导出和导入实用程序

您可以通过以下任何方法来调用导出和导入实用程序并指定参数：

- 命令行输入
- 交互式提示
- 参数文件
- Oracle Enterprise Manager 数据管理向导

要使用导出实用程序，必须对 Oracle 数据库拥有 CREATE SESSION 权限。要导出由另一个用户拥有的表，必须已启用了 EXP_FULL_DATABASE 角色。该角色授予所有 DBA。如果您不具有 EXP_FULL_DATABASE 角色中所包含的系统权限，则无法导出包含在另一个用户的方案中的对象。

要使用导入实用程序，需要具有 CREATE SESSION 权限以便登录到 Oracle 数据库服务器。此权限属于在创建数据库期间建立的 CONNECT 角色。

即使您没有创建导出文件，也可以执行导入。但是，如果导出文件是由其他人而不是您创建的，则只当您具有 IMP_FULL_DATABASE 角色的权限时，才可以导入该文件。

注：许多选项只能通过命令行界面才能使用。不过，您可以将参数文件与命令行一起使用。

导出模式

表模式	用户模式	表空间模式	完全数据库模式
表定义	表定义	表定义	表定义
表数据（所有行或选定的行）	表数据		表数据
所有者的表授权	所有者的授权	授权	授权
所有者的表索引	所有者的索引	索引	索引
表约束	表约束	表约束 触发器	表约束

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

导出模式

表模式

表模式可以导出用户方案中的指定的表，而不是导出所有表。已授权的用户可以导出其他用户所拥有的指定的表。

用户模式

用户模式可以导出某个用户方案的所有对象。已授权的用户可以导出一组指定用户的方案中的所有对象。此模式可用于执行完全数据库导出。

表空间模式

表空间模式可用于导出指定表空间中的所有表。

使用可移动的表空间功能，可以将一组表空间从一个 Oracle 数据库移动到另一个 Oracle 数据库。

利用可移动的表空间来移动数据，其速度比对相同数据执行导入/导出要快得多，原因是传输表空间只需要复制数据文件，然后并入表空间结构信息即可。您还可以使用可移动的表空间来移动索引数据，从而避免在导入表数据时必须重建索引。

导出模式（续）

完全数据库模式

完全数据库模式可以导出所有数据库对象，SYS 方案中的对象除外。只有已授权的用户才能在此模式中执行导出。

注：这是部分处理。有关模式和对象的完全处理，请参阅 “Oracle Utilities Guide”

调用导出实用程序

- 语法:

```
exp keyword = value, value2, ... ,valuen
```

- 示例:

```
exp hr/hr TABLES=employees,departments  
rows=y file=exp1.dmp
```

```
exp system/manager OWNER=hr direct=y  
file=expdat.dmp
```

```
exp \'username/password AS SYSDBA\'  
TRANSPORT_TABLESPACE=y  
TABLESPACES=ts_emp log=ts_emp.log
```

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

调用导出实用程序

命令行导出

可以使用导出实用程序的命令行模式将数据库数据复制到某个操作系统文件中。只有导入实用程序才能读取该文件。

示例

创建名为 exp1.dmp 的导出文件，其中包含 HR 方案中的表 EMPLOYEES 和 DEPARTMENTS（包括表行）：

```
exp hr/hr tables=employees,departments rows=y file=exp1.dmp
```

注：表名称中可包含任意数量的 '%' 模式匹配字符，其中每个字符可将表名称中的零或多个字符与数据库中的表对象相匹配。

创建一个名为 expdat.dmp 的快速导出文件，其中包含 HR 方案的所有对象：

```
exp system/manager owner=hr direct=y
```

注：缺省的导出文件名为 expdat.dmp。

调用导出实用程序（续）

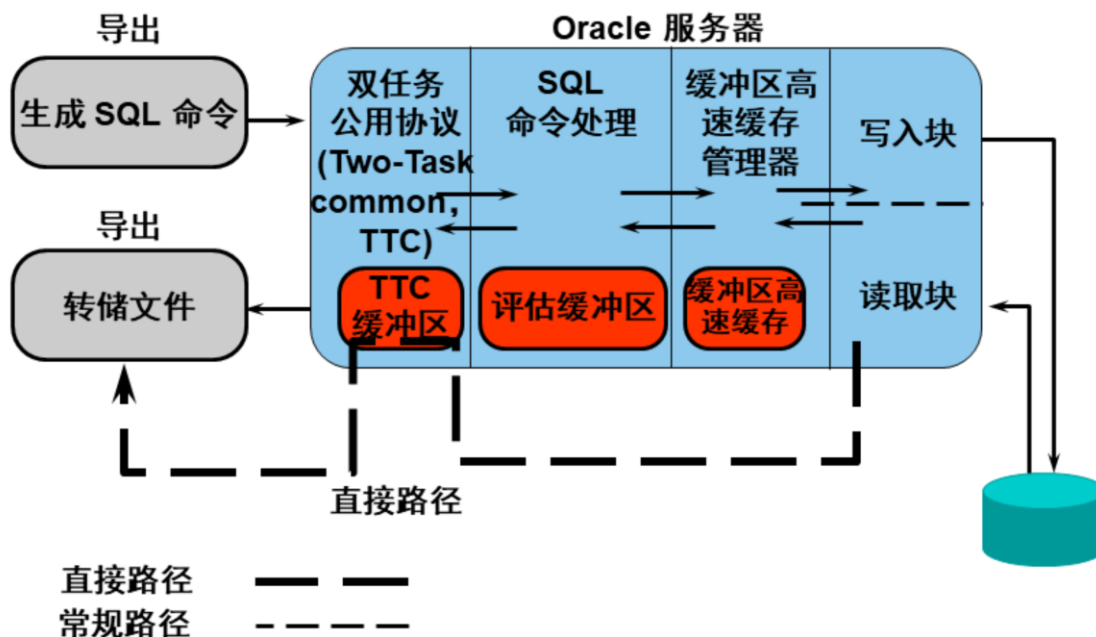
创建一个名为 expdat.dmp 的导出文件，该文件中包含属于 TS_EMPLOYEES 表空间的所有的对象定义，并生成一个名为 ts_employees.log 的日志文件：

```
exp \ 'username/password AS SYSDBA\ ' TRANSPORT_TABLESPACE=y\  
TABLESPACES=ts_employees LOG=ts_employees.log
```

导出参数

参数	说明	缺省值
BUFFER	指定用于提取行的缓冲区的大小 (以字节为单位)	取决于操作系统
COMPRESS	指定导出实用程序如何管理表数据的初始区	Y
CONSISTENT	指定导出实用程序是否使用 SET TRANSACTION READ ONLY	Y
DIRECT	指定使用直接路径	N
FEEDBACK	指定为已导出的 n 行显示时间段形式的进度表。	0
FILE	指定导出文件的名称	expdat.dmp
FULL	指示以完全数据库模式导出	N
HELP	显示参数的说明	N
INDEXES	指定是否导出索引	Y
LOG	指定接收信息性消息和错误消息的文件名	无
OWNER	指示以用户模式导出, 并列出要导出的对象所属的用户	无
PARFILE	指定包含有导出参数列表的文件	无
RECORDLENGTH	指定文件记录长度 (以字节为单位)	取决于操作系统
ROWS	指定是否导出表数据的行	Y
TABLES	指定以表模式导出, 并列出要导出的表名称、分区和子分区名称	无
TABLESPACES	指定导出表空间中的所有表	无
TRANSPORT_ TABLESPACE	启用导出可移动的表空间元数据	N
USERID	指定执行导出操作的用户的用户名/ 口令	无

直接路径导出概念



ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

直接路径导出概念

使用直接路径功能可以大大加快提取数据的速度。如果指定了 `DIRECT=Y` 参数，导出实用程序将直接从数据层读取，而不必通过 SQL 命令处理层。

直接路径导出机制

- 直接导出模式可以通过指定 `DIRECT=Y` 参数来设置。
- 直接路径导出不与例程的其它资源竞争。
- 在直接读取模式中，它将数据库的数据块读入由会话使用的一个专用区域中。
- 行将被直接传送到双任务通信协议 (Two-Task Common, TTC) 缓冲区以供传输。
- TTC 协议缓冲区中数据的格式是导出实用程序预期的格式。

直接路径导出功能

- 导出类型将显示在屏幕输出、导出转储文件和日志文件中。
- 数据已经是导出实用程序所预期的格式，从而避免了不必要的数据转换。
- 使用优化的 **SQL SELECT** 语句。

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

直接路径导出

导出实用程序的直接路径选项引入了一些有别于常规路径导出的功能。

直接路径功能

- 导出的类型将显示在屏幕输出、导出转储文件和以 LOG 参数指定的日志文件中。
- 数据已经是导出实用程序所预期的格式，因此避免了不必要的数据转换。数据将传送到导出客户机，该客户机随后将数据写入导出文件。
- 直接路径导出使用经过优化的 **SELECT * FROM** 表，而不带任何谓词。

注：导出转储文件中的列数据和说明的格式与常规路径导出的相应格式不同。

直接路径导出的限制

- 直接路径选项不能交互式调用。
- 客户端和服务端端的字符集必须相同。
- **BUFFER** 参数不起作用。
- 不能使用直接路径选项来导出包含 **LOB**、**BFILE**、**REF** 的行或对象类型。

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

直接路径的限制

导出实用程序的直接路径选项具有一些有别于常规路径导出的限制。

- 直接路径导出功能不能通过使用交互式 EXP 会话来进行调用。
- 在使用直接路径选项时，客户端的字符集必须与服务器端的字符集相匹配。可以使用环境变量 **NLS_LANG** 设置为与服务器端相同的字符集。
- 导出实用程序的 **BUFFER** 参数对直接路径导出不起作用；只有常规路径选项使用该参数。
- 不能使用直接路径导出功能来导出包含有 **LOB**、**BFILE**、**REF** 数据类型的行或对象类型列（包括 **VARRAY** 列和嵌套表）。只导出用来创建表的数据定义，而不导出数据。

指定直接路径导出

- 使用导出命令的命令行参数：

```
exp userid=hr/hr full=y direct=y
```

- 使用参数文件中的关键字：

```
exp parfile=<Parameter file>
```

参数文件

..... (其它参数)

DIRECT=Y

..... (其它参数)

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

指定直接路径导出

在可以使用直接路径导出之前，必须确保已运行了 catexp.sql 脚本。它位于 \$HOME/rdbms/admin/ 目录下。

命令行选项

可以在操作系统提示符下使用 DIRECT 命令行参数来调用直接路径导出。

```
$ exp user=hr/hr full=y direct=y
```

参数文件

exp_par.txt 是参数文件的一个示例：

```
USERID=hr/hr
```

```
TABLES=employees,departments
```

```
FILE=exp_one.dmp
```

```
DIRECT=Y
```

从操作系统提示符下使用参数文件调用导出实用程序：

```
$ exp parfile=exp_param.txt
```

使用导入实用程序进行恢复

- 创建表定义
- 从有效的导出文件中提取数据
- 从完全或累积导出文件中导入
- 从用户错误故障中恢复

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

导入实用程序

导入实用程序从导出转储文件中读取对象定义和表数据。它将数据对象插入 Oracle 数据库。导入功能包括：

- 创建表定义（因为表定义存储在导出文件中）。选择导入不带行的数据将只创建表定义。
- 从有效的导出文件中提取数据，方法是使用表模式、用户模式、表空间模式或完全导入模式
- 从完全或累积导出文件中导入数据
- 使用前面所述的方法之一从用户操作失败而导致的错误中（如意外删除或截断了某个表）恢复

导入模式

模式	说明
表模式	将指定的表导入方案中。
用户模式	导入属于某个方案的所有对象
表空间模式	导入表空间中包含的所有的对象定义
完全数据库模式	从导出文件中导入所有对象

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

导入模式

表模式

表模式可以导入用户方案中所有指定的表，而不是导入所有表。已授权的用户可以导入其他用户拥有的指定的表。

用户模式

用户模式可以导入某个用户方案的所有对象。已授权的用户可以导入一组指定用户的方案中的所有对象。

表空间模式

表空间模式允许已授权的用户将一组表空间从一个 Oracle 数据库移动到另一个 Oracle 数据库。

完全数据库模式

完全数据库模式可以导入所有数据库对象，SYS 方案中的对象除外。只有已授权的用户才能以此模式执行导入。

调用导入实用程序

- 语法:

```
imp keyword = value or keyword = value,  
value2, ... value n
```

- 示例:

```
imp hr/hr TABLES=employees,departments  
rows=y file=exp1.dmp
```

```
imp system/manager FROMUSER=hr file=exp2.dmp
```

```
imp \'username/password AS SYSDBA\'  
TRANSPORT_TABLESPACE=y  
TABLESPACES=ts_employees
```

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

调用导入实用程序

导入示例

使用名为 exp1.dmp 的导出文件，将表 EMPLOYEES 和 DEPARTMENTS（包括表行）导入 HR 方案中。

```
$ imp hr/hr tables=employees,departments rows=y file=exp1.dmp
```

使用名为 exp2.dmp 的导出文件，导入所有属于 HR 方案的对象（包括表行）。

```
$ imp system/manager FROMUSER=hr file=exp2.dmp
```

使用名为 expdat.dmp 的导出文件，导入所有属于表空间 TS_EMPLOYEES 的对象定义。

```
$ imp \'username/password AS SYSDBA\' \  
TRANSPORT_TABLESPACE=y TABLESPACES=ts_employees
```

如果未指定文件参数，导入实用程序会查找缺省的文件 expdat.dmp。

注： 命令行模式选项与交互模式选项相似，但却提供更多的功能。

导入参数

参数	说明	缺省值
BUFFER	指定用来传送数据行的缓冲区大小 (以字节为单位)	取决于操作系统
DATAFILES	列出要传输到数据库中的数据文件	无
DESTROY	指定是否应重复使用构成数据库的现有数据文件	N
FEEDBACK	为已导入的 n 行显示时间段形式的进度表	0
FILE	指定要导入的导出文件的名称	expdat.dmp
FROMUSER	要导入的方案的列表	无
FULL	指定是否导入整个导出文件	N
HELP	显示导入参数的说明	N
IGNORE	指定应如何处理对象创建错误	N
INDEXES	指定是否导入索引	Y
INDEXFILE	指定接收索引创建语句的文件	无
LOG	指定接收信息性消息和错误消息的文件	无
PARFILE	指定包含有导入参数列表的文件	无
RECORDLENGTH	指定文件记录长度 (以字节为单位)	取决于操作系统
ROWS	指定是否导入表数据的行	Y
TABLES	指定以表模式导入, 并列出要导入的表名称、分区和子分区名称	无
TABLESPACES	指定要传输到数据库中的表空间列表	无
TOUSER	指定其方案要作为导入目标的用户名列表	无
TRANSPORT_ TABLESPACE	从导出文件中导入可移动的表空间元数据	N
USERID	指定执行导入操作的用户的用户名/ 口令	无

作为 SYSDBA 调用导入实用程序

- 在下列条件下，您需要以 SYSDBA 用户身份调用导入实用程序：
 - 应 Oracle 技术支持的要求
 - 在导入可移动的表空间集时
- 要以 SYSDBA 用户身份来调用导入实用程序，请执行以下命令：

```
imp \ 'username/password AS SYSDBA\ '
```

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

作为 SYSDBA 导入

SYSDBA 在内部使用并具有专门功能。它的行为与一般用户的行为不同。因此，通常应不需要以 SYSDBA 用户身份来调用导入实用程序，除了以下两种情况：

- 应 Oracle 技术支持的要求
- 在导入可移动的表空间集时

要以 SYSDBA 用户身份来调用导入实用程序，请使用下面的语法，并添加任何所需的参数或参数文件名：

```
imp \ 'username/password AS SYSDBA\ '
```

或者，也可以使用：

```
imp \ 'username/password@instance AS SYSDBA\ '
```

如果遗漏了用户名或口令，导入实用程序将提示您输入它。在该示例中，完整的连接字符串由引号和反斜杠括起来。这是因为字符串 AS SYSDBA 中包含一个空格；在这种情况下，大多数操作系统都要求将整个连接字符串放在引号内，或用某种方法将其标记为一个字。此外，某些操作系统还要求命令行中的引号前应具有转义符。在该示例中，反斜杠即用作转义符。如果没有反斜杠，导入实用程序所使用的命令行语法分析器将无法理解引号，而会在调用导入实用程序前删除它们。

导入进程序列

1. 创建新表
2. 导入数据
3. 建立索引
4. 导入触发器
5. 对新表启用完整性约束
6. 建立所有位图、函数和/或域索引

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

导入进程序列

在导入一个表时，将读取导出文件，并按照以下顺序创建表和数据：

1. 创建新表
2. 导入数据
3. 建立索引
4. 导入触发器
5. 对新表启用完整性约束
6. 建立所有位图、函数和/或域索引

该顺序可防止数据由于表的导入顺序而被拒绝。该顺序还可防止对同一数据两次触发冗余触发器（一次在初始插入时，一次在导入过程中）。

如果不导入某个用户拥有的所有对象，那么导入表的顺序将非常重要。例如，如果具有外键的表对具有主键的表进行引用检查，而且先导入外键表，则在启用了约束条件的情况下，将拒绝所有引用该尚未导入的主键的行。在完全数据库导出中不存在该问题。

全球化支持注意事项

- 导出文件确定该文件中字符数据所使用的字符编码方案。
- 导入实用程序将数据转换为其主机系统的字符集。
- 多字节字符集导出文件必须导入到具有相同特征的系统
中。

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.

全球化支持注意事项

将数据从使用一种字符集的 Oracle 数据库移动到另一个使用其它字符集的数据库时，请确保必须对数据进行相应的转换。可以将 NLS_LANG 环境变量设置为要从其中导出数据的数据库的字符集定义，以便进行数据转换。如果没有正确设置该变量，数据中的字符可能会出现意外转换，从而导致数据丢失。

示例

将 7 位 ASCII 字符集（如美国英语）转换为 8 位字符集（如丹麦语）时，因为所有字符在丹麦语字母表中均有对应的字符，所以无需转换。

而将 8 位 ASCII 字符集（如丹麦语）转换为 7 位字符集（如美国英语）时，会将美国英语字母表中找不到的特别的丹麦语字符转换为问号 (?)。在这种情况下，使用问号代替未知的丹麦语字符是一个合适的结果。

在 8 位至 8 位的数据移动中，是否丢失字符取决于用来输入数据的语言的特性。例如，西班牙语字母表中的某些字母在丹麦语字母表中没有，所以将数据从西班牙语数据库移动到丹麦语数据库时可能会需要数据转换，从而可能会丢失这些字符。

小结

在这一课中，您应该能够掌握：

- 描述导出和导入实用程序的使用
- 说明导出和导入的概念与结构
- 执行简单的导出和导入操作
- 列出使用导出和导入操作的原则

ORACLE

Copyright © Oracle Corporation, 2001. All rights reserved.