

Oracle GoldenGate

数据库同步技术

Oracle GoldenGate TDM 是一种基于软件的数据复制方式，它从数据库的日志解析数据的变化。

- *Oracle Data Integrator*
- *Oracle GoldenGate 的技术原理*
- *利用 GoldenGate 同步 SQL Sever*

2000 数据



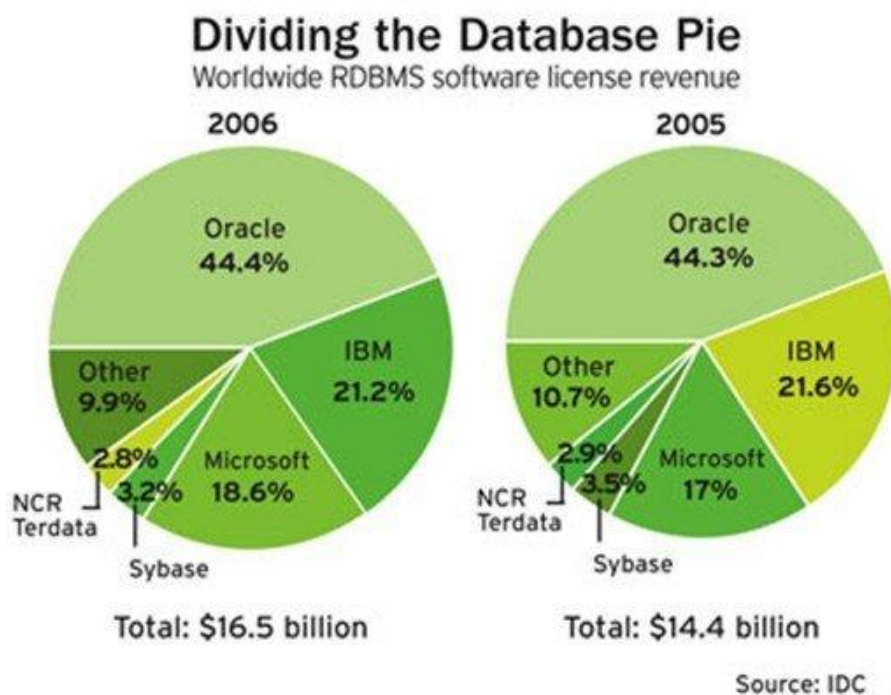
Oracle GoldenGate

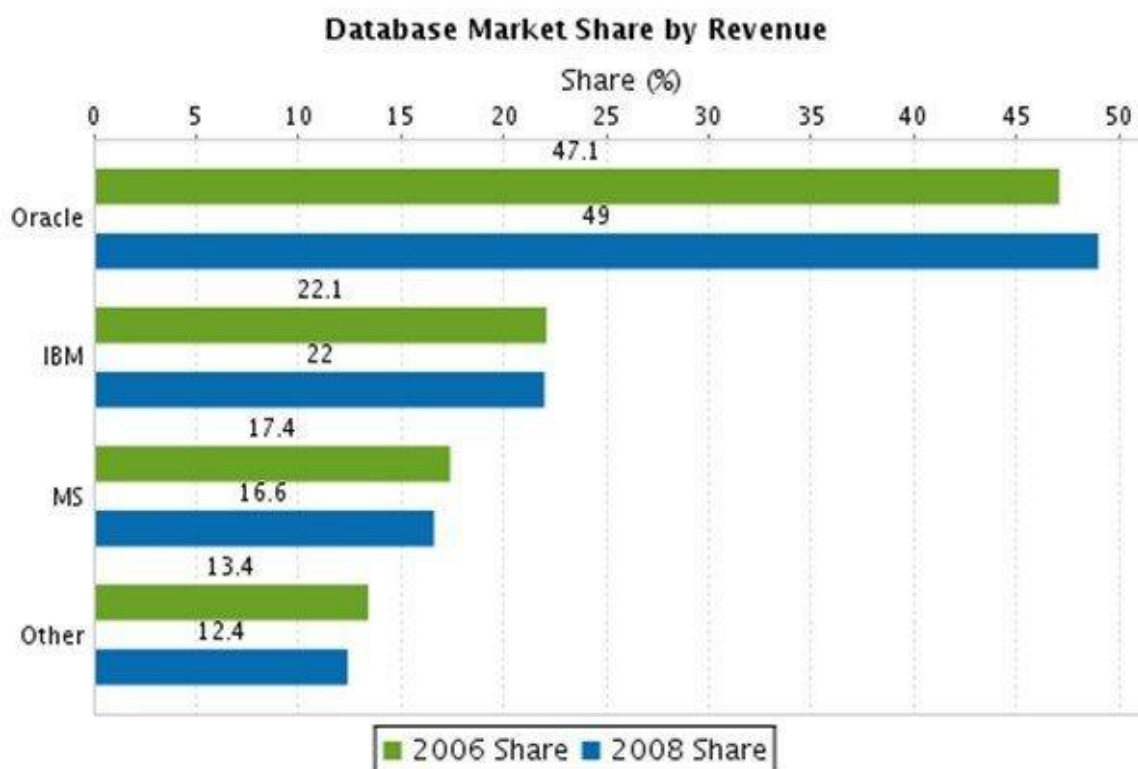
数据库同步技术

Oracle GoldenGate 是一款实时访问、基于日志变化捕捉数据，并且在异构平台之间进行数据传输的产品。利用这个技术，GoldenGate 能够实现低延迟、高性能、低影响进行连续的数据集成。——赵宇

开始切入正题之前，我们先来看一项数据库市场的调查：权威调查机构的报告显示，数据库市场在近几年分布占据主导地位的还是 Oracle、DB2、SQL Server。现在随着企业信息

化水平的不断提高，很多企业也都上很多信息化的系统，比如 OA 系统、订单系统、网站、ERP、数据仓库、计费系统等系统，在证券、银行、电信、保险、网站等行业更是多个数据库系统并存的情况。数据库多种多样、版本又不统一，这样就给企业在做数据分析以及汇总带来很大的难度。通常客户会选择自己写脚本、开发程序来做数据的同步，往往效率不高，程序还需要经常维护。SQL Server 在中小企业、证券市场占比较大的比重，我们这里就举例 SQL Server 之间的数据库同步。





如果不通过开发程序代码来完成 SQL Server 2000/2005 的数据同步到 SQL Server 2000/2005 数据库中，有哪些方案可以选择呢？SQL Server 本身就提供了很多解决方案，比如大家很熟悉的 Microsoft Snapshot Replication，Microsoft Transactional Replication，Microsoft Merge Replication。但是这些方案在实时数据捕捉、数据同步灵活性以及双向复制方面都不够完善，我们在后期电子书中也会介绍。Oracle 公司在过去 3 年收购了 2 款强大的产品，Oracle Data Integrator 和 Oracle Goldengate。

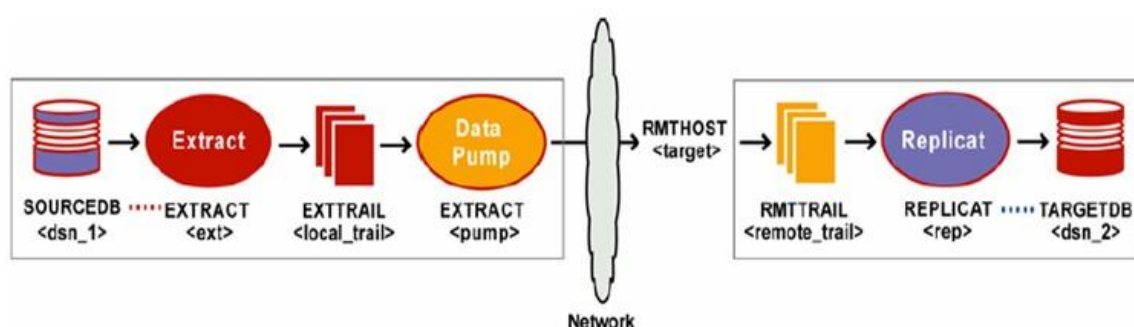
Oracle Data Integrator

Oracle Data Integrator 数据集成器是一个全面的数据集成平台，它能满足所有数据集成要求，涉及领域包括高容量、高性能批处理、事件驱动的少量传送集成过程以及支持 SOA 的数据服务。ODI 集成了很多数据库的功能模块以及 java 开发完整的模块，可以利用 JDBC 协助我们完成 MSSQL 到 Oracle 的同步，但是 ODI 强大的功能在于可以对数据进行提取、过滤、转换、加载，比传统的 ETL 工具更完善，而且可以实现 workflow、流程控制等。在这期杂志，我们主要是利用 Oracle Goldengate 这个产品来做异构数据库、异构平台间的数据同步。

Oracle Goldengate 是一款实时访问、基于日志变化捕捉数据，并且在异构平台之间进行数据传输的产品。利用这个技术，Goldengate 能够实现低延迟、高性能、低影响进行连续的数据集成。

Oracle GoldenGate 技术原理

GoldenGate TDM（交易数据管理）软件是一种基于日志的结构化数据复制软件，它通过解析源数据库在线日志或归档日志获得数据的增删改变化，再将这些变化应用到目标数据库，实现源数据库与目标数据库同步、双活。GoldenGate TDM 软件可以在异构的 IT 基础结构（包括几乎所有常用操作系统平台和数据库平台）之间实现大量数据亚秒一级的实时复制，其复制过程简图如下：



如上图所示，GoldenGate TDM 的数据复制过程如下：

利用捕捉进程 (Extract Process) 在源系统端读取 Online Redo Log 或 Archive Log，然后进行解析，只提取其中数据的变化如增、删、改操作，并将相关信息转换为 GoldenGate TDM 自定义的中间格式存放在队列文件 (trail file) 中。再利用传送进程将队列文件通过 TCP/IP 传送到目标系统。捕捉进程在每次读完 log 中的数据变化并在数据传送到目标系统后，会写检查点，记录当前完成捕捉的 log 位置，检查点的存在可以使捕捉进程在中止并恢复后可从检查点位置继续复制；

目标系统接受数据变化并缓存到 GoldenGate TDM 队列当中，队列为系列临时存储数据变化的文件，等待投递进程读取数据；

GoldenGate TDM 投递进程(replicat process)从队列中读取数据变化并创建对应的 SQL 语句，通过数据库的本地接口执行，提交到数据库成功后更新自己的检查点，记录已经完成复制的位置，数据的复制过程最终完成。

由此可见，GoldenGate TDM 是一种基于软件的数据复制方式，它从数据库的日志解析数据的变化（数据量只有日志的四分之一左右）。GoldenGate TDM 将数据变化转化为自己的格式，直接通过 TCP/IP 网络传输，无需依赖于数据库自身的传递方式，而且可以通过高达 10:1 的压缩率对数据进行压缩，可以大大降低带宽需求。在目标端，GoldenGate TDM 可以通过交易重组，分批加载等技术手段大大加快数据投递的速度和效率，降低目标系统的资源占用，可以在亚秒级实现大量数据的复制，并且目标端数据库是活动的。

下面详细介绍利用 Goldengate 同步 SQL Sever 2000 到 SQL Server 2000 表数据的具体实现。

利用 GoldenGate 同步 SQL Sever 2000 数据

测试环境: Windows XP

Goldengate V10.4

源数据库：MSSQL 2000 db01

目标数据库：MSSQL 2000 db02

测试用户：Db01 库：ggstrn01

Db02 库：ggstrn02

```
C:\Documents and Settings\goldengate>cd c:\gg-sql

Oracle GoldenGate Command Interpreter for ODBC

Version 10.4.0.19 Build 002

Windows (optimized), Microsoft SQL Server on Sep 21 2009 09:10:43

Copyright (C) 1995, 2009, Oracle and/or its affiliates. All rights reserved.

GGSCI (GGSEDU50) 1> edit params ./GLOBALS

checkpointtable ggusr.checkpoint

MGRSERVNAME GGMGRSQLSERVER

GGSCI (GGSEDU50) 3> EXIT

C:\gg-sql>install ADDSERVICE AUTOSTART

Service 'GGMGRSQLSERVER' created.

Install program terminated normally.
```

GLOBALS 是 Goldengate 的全局参数文件，编辑好以后呢，需要退出 GGSCI 才能生效。然后验证是否 Windows 生产了相应的服务。


```
C:\gg-sql>sc query GGMGRSQLSERVER
```

```
SERVICE_NAME: GGMGRSQLSERVER
```

```
        TYPE               : 10  WIN32_OWN_PROCESS
```

```
        STATE               : 1  STOPPED
```

```
(NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
```

```
        WIN32_EXIT_CODE      : 1077      (0x435)
```

```
        SERVICE_EXIT_CODE   : 0  (0x0)
```

```
        CHECKPOINT          : 0x0
```

```
        WAIT_HINT           : 0x0
```

在利用 Goldengate 做数据同步之前，需要配置数据库的相关配置，具体如下：

- **指定数据源行的键**

指定行的唯一标识

Goldengate 推荐能够在源库的行级别有唯一的标识，比如主键、唯一键，这样在目标数据库做删除、修改时能及时定位相关的行数据。如果没有主键、唯一键，Goldengate 会把所有的数据都作为 Where 条件进行数据操作的判断条件。

Goldengate 会决定用哪种行唯一标识

GoldenGate 选择行唯一标识的优先级如下：

1. Primary key 主键；
2. 第一个不包含 timestamp 的唯一键；
3. 也可以用 KEYCOLS 在 Extract TABLE 参数或者 Replicat MAP 参数中指定可以判断唯一行的列。这个指定的列会覆盖 Goldengate 自己寻找到的唯一标识。

注意 如果没有主键、唯一键也没有用 KEYCOLS 指定，GoldenGate 会在进程报告中提示消息，表明会用所有列的信息做唯一标识。

失效触发器和级联删除的约束

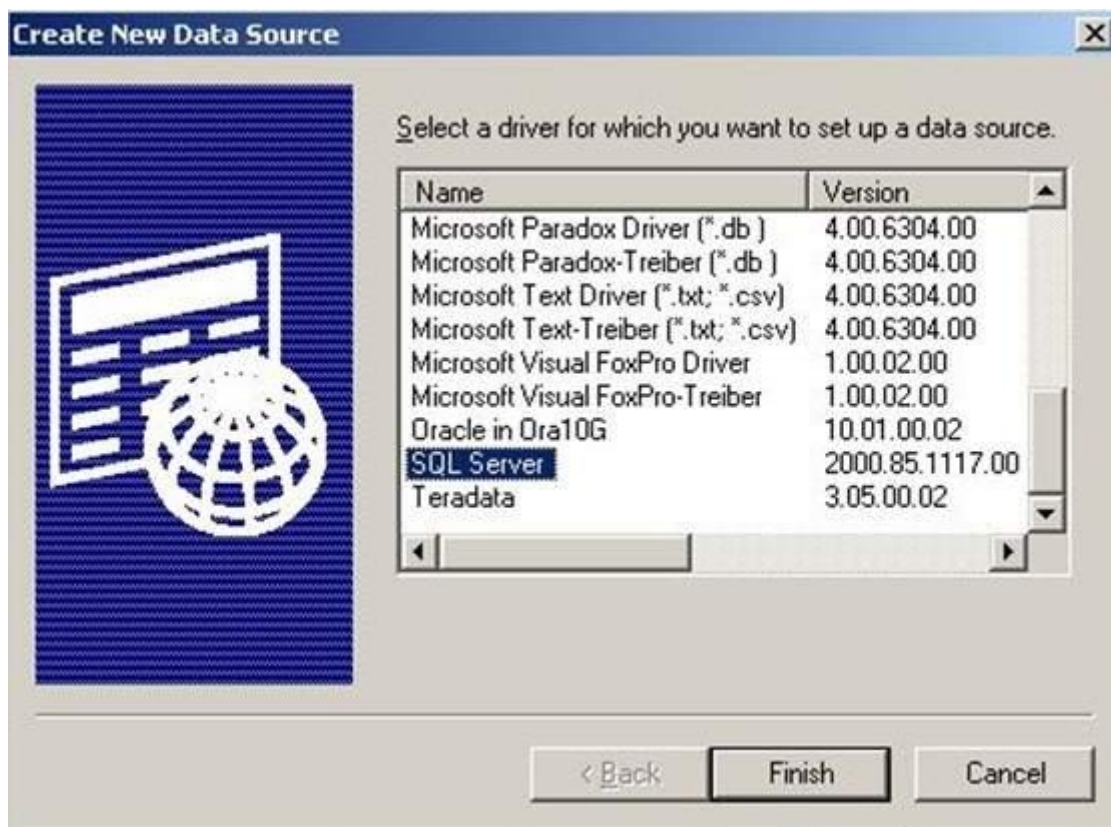
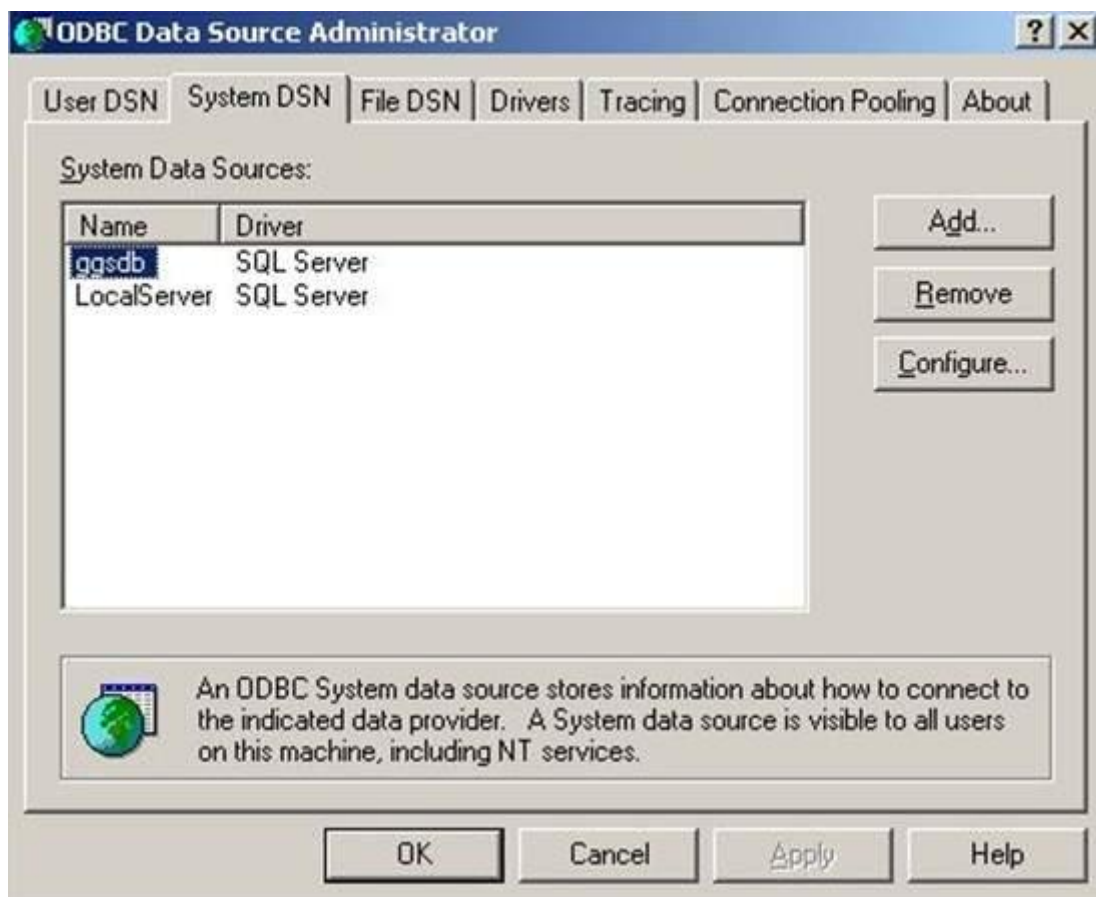
在目标数据库，也就是本次测试的 oracle 数据库中停掉触发器和级联删除的约束,或者让 Goldengate 忽略他们产生的变化。

下面要配置 ODBC 连接 MSSQL 数据库，由于可以通过 ODBC 连接，ODBC 也可以配置远程的 MSSQL 数据库。

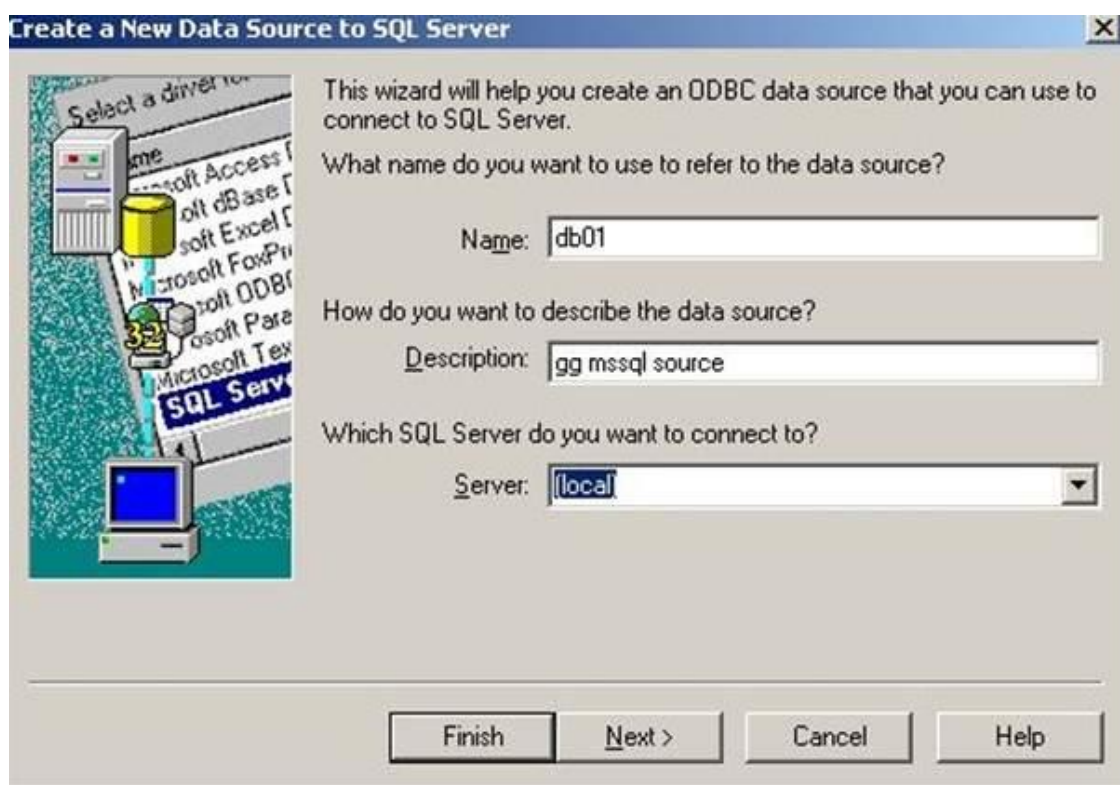
● 配置 ODBC 连接

创建一个 SQL Server DSN

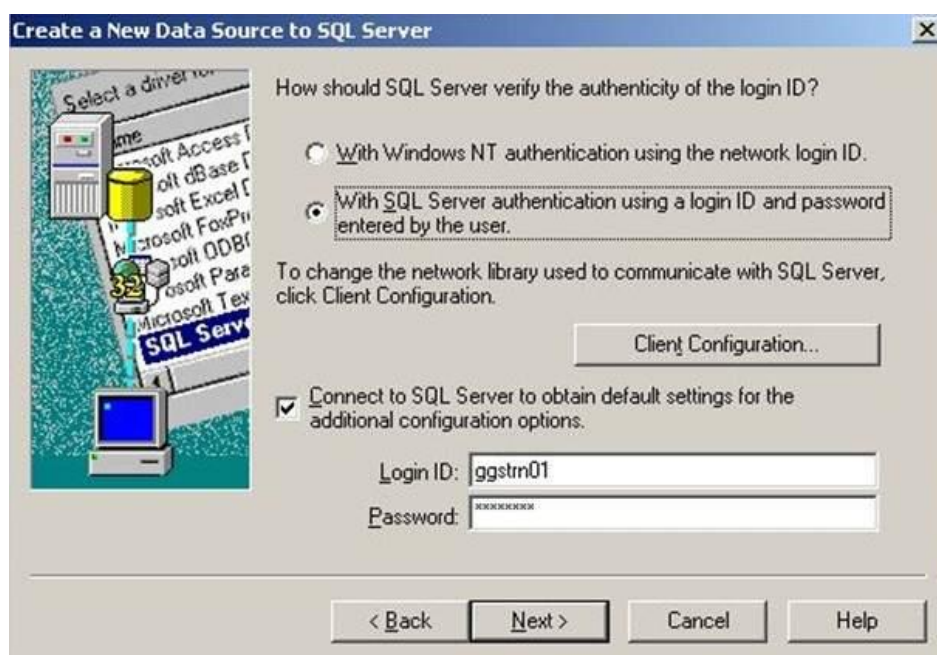
控制面板>管理工具 >数据源(ODBC)



选择 MSSQL 的数据服务器，指定源数据库的名称 db01



在 MSSQL 里创建一个专门为 Goldengate 使用的用户，输入用户名和密码

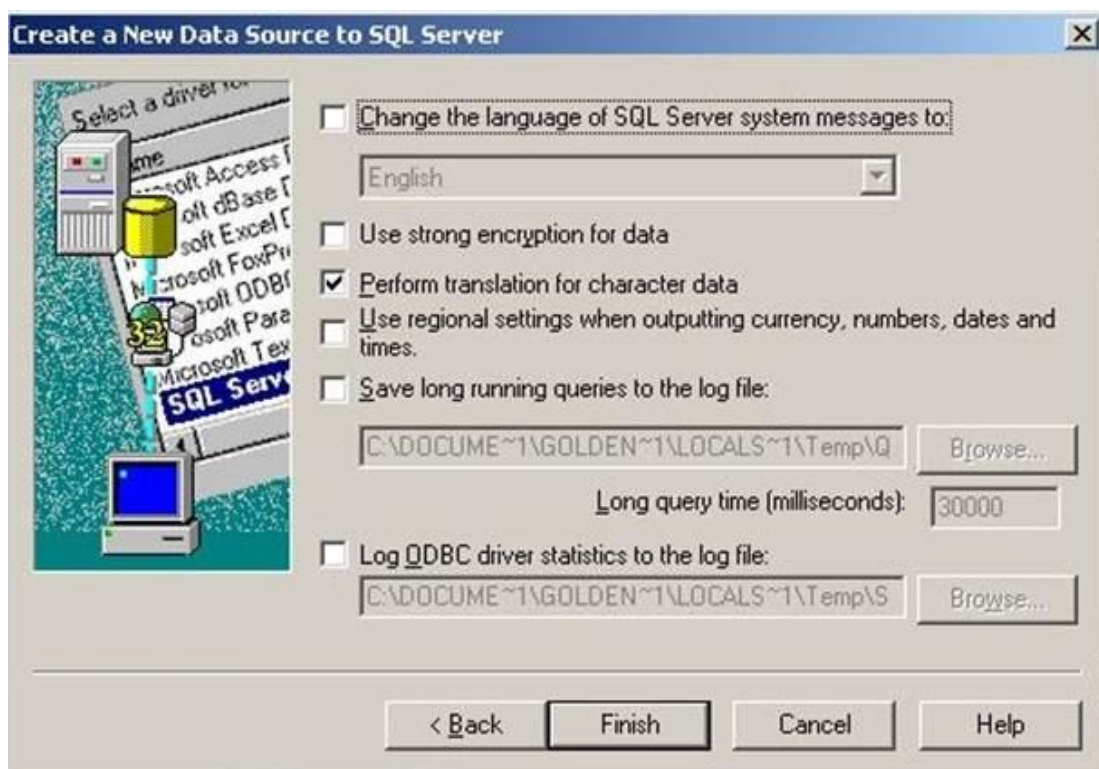


在这里，我们是把为 goldengate 分配的用户存放了数据，实际生产环境中，为 goldengate 创建一个用户用于提取日志的操作，赋予相应管理权限。而数据通常存放在其他的用户下面。

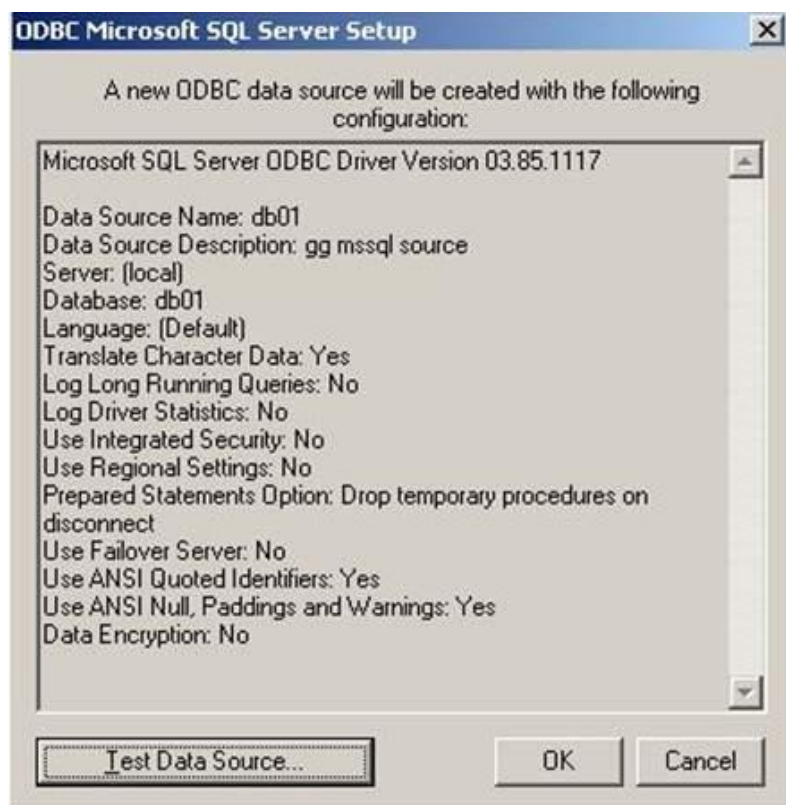
然后选择默认的要提取数据的数据库



默认设置



测试数据库的连接





- **配置 SQL Server 2000 的日志模式**

为了让 GoldenGate 从 SQL Server 源系统提取日志的信息，需要做下面的设置：

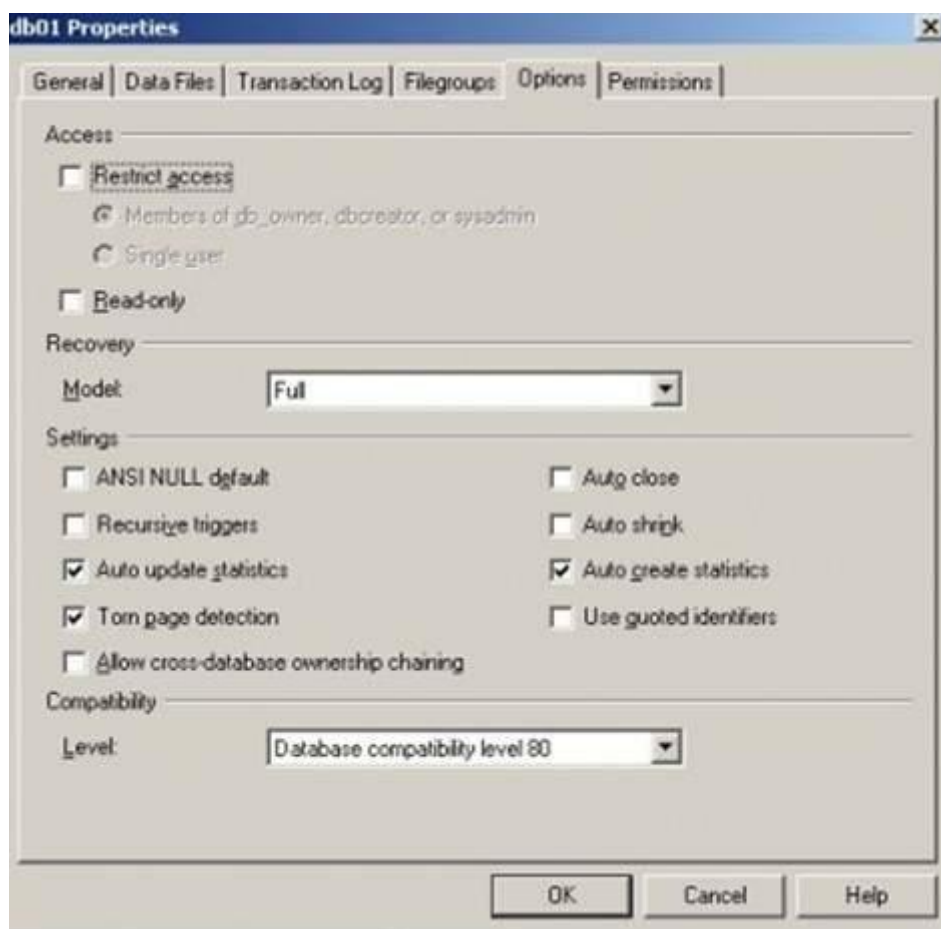
- 设置数据库的日志模式为 full recovery model. GoldenGate 要求关闭 log truncation 和 non-logged bulk copy.
- 在启动 Goldengate 之前至少要做一次全数据库备份。

验证或者设置恢复模式

1、用 SQL Server 2000 的企业管理器或者 for SQL Server 2005/2000

的 SQL Server Management Studio 进入 SQL SERVER 的实例；

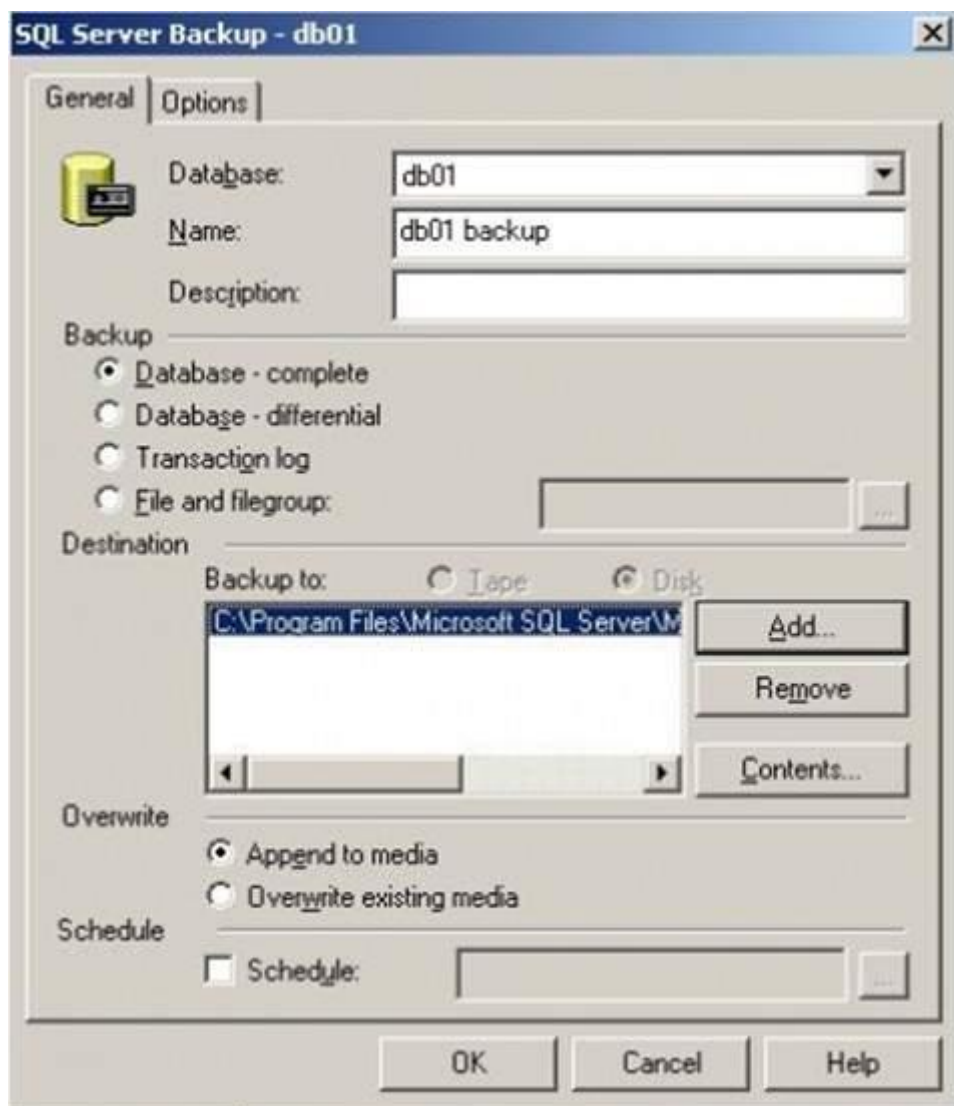
- 2、展开数据库的文件夹；
- 3、右键单击源数据库, 然后选择属性；
- 4、选择 option tab 页面。



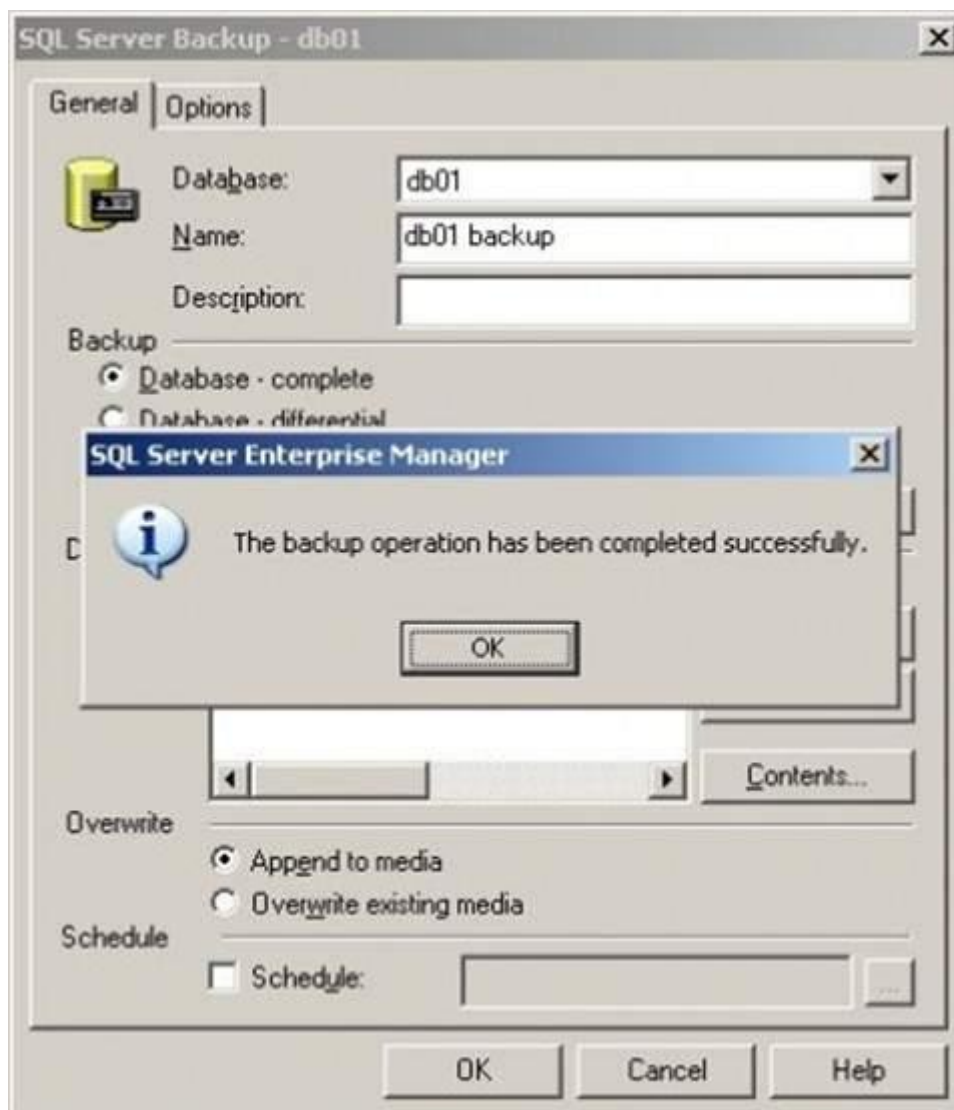
完全备份数据库

- 1、右键点击数据库的名字, 选择 All Tasks > Backup Database.
- 2、Select Database – Complete. 这个选项做全数据库备份，并且确保

GoldenGate 启动时不会有任何的事务丢失。



然后选择数据库备份的位置。



备份成功提示。

在前面的工作完成以后，数据库的准备工作就已经基本完成了，下面我们就开始直接配置 Goldengate 针对源数据库 SQL Server 的配置。

- **启用扩展日志（附加日志）**

启用扩展日志或附加日志是可以在数据库里完成，也可以在 Goldengate 里完成，我们主要是介绍怎样配置 Goldengate，Goldengate 安装就不在这里介绍，具体安装配置，参考：

http://download.oracle.com/docs/cd/E15881_01/doc.104/gg_mss_inst_v104.pdf

我们在后期电子书中也会详细介绍。

我们可以在 Goldengate 里面利用 ADD TRANDATA 命令启用扩展日志，让提取进程 Extract 捕获用以在目标库重构 SQL 所需要的日志信息。扩展日志信息比 SQL Server 默认日志包含了更多数据的信息。这个命令要求对所有需要复制的表执行。

ADD TRANDATA 实现下面功能：

如果源数据库是 SQL Server 2000, ADD TRANDATA 设置一个标识在 sysobjects 表，强制 SQL Server 日志记录前镜像和后镜像信息，也就是变化前的数据和变化后的数据。在 Goldengate 启用扩展日志的命令：

```
C:\gg-sql>ggsci  
  
Oracle GoldenGate Command Interpreter for ODBC  
  
Version 10.4.0.19 Build 002
```

Version 10.4.0.19 Build 002

Windows (optimized), Microsoft SQL Server on Sep 21 2009 09:10:43

Copyright (C) 1995, 2009, Oracle and/or its affiliates. All rights reserved.

GGSCI (GGSEDU50) 2> dblogin sourcedb db01, userid ggstrn01, password
ggstrn01

Successfully logged into database.

○ SOURCEDB <DSN> is the name of the data source.

○ USERID <user>, PASSWORD <password> is the Extract login and password, if
Extract uses

SQL Server authentication. Can be the Extract user or a member of an account in
the System Administrators or Server Administrators fixed server role.

GGSCI (GGSEDU50) 5> add trandata ggstrn01.*

Logging of supplemental log data is enabled for table ggstrn01.emp

注释：

指定在线和备份日志的位置，指定一个可选的在线日志位置。改变提取进程
Extract 到指定的位置抓取在线日志用下面参数 TRANLOGOPTIONS 结合参数
ALTONLINELOGS，可以指定多个日志位置。

- **指定备份日志的可选位置**

默认情况下,如果在线日志不包括正在处理的事务,提取进程 Extract 会直接读备份的日志。如果提取进程延迟很大,那么就会读取备份的日志,在处理以后,就不需要读取这些日志。

Extract 进程通常会直接到 SQL Server 的默认位置去读备份日志.如果你已经改变了默认的存储位置,你必须在 EXTRACT 进程中指定参数 TRANLOGOPTIONS 以及 ALTARCHIVELOGDEST 选项,告诉 Extract 去哪里读取日志。

- **在 SQL Server 源数据库创建提取进程 Extract Group、DUMP Group**

```
GGSCI (GGSEDU50) 8> ADD EXTRACT EXT_MSS, TRANLOG, BEGIN NOW
EXTRACT added.

GGSCI (GGSEDU50) 11> ADD EXTTRAIL ./dirdat/et, EXTRACT EXT_MSS,
MEGABYTES 50
EXTTRAIL added.

GGSCI (GGSEDU50) 10> ADD EXTRACT DUMP_MSS, EXTTRAILSOURCE ./dirdat/et
EXTRACT added.

GGSCI (GGSEDU50) 39> add RMTTRAIL c:/gg-sql/dirtrail/rt, EXTRACT DUMP_MSS
RMTTRAIL added.
```

- **因为进程创建以后，需要指定进程的参数 Extract Parameter**

```
GGSCI (GGSEDU50) 18> view param ext_mss

extract ext_mss

dboptions NOCATALOGCONNECT, FETCHBATCHSIZE 1500,

sourcedb db01, userid ggstrn01, password ggstrn01

--tranlogoptions ALTARCHIVELOGDEST , ALTONLINELOGS ("c:\SQL
Server\Data\NorthWnd1.ldf"), EXCLUDETRANS ggs_repl, QUERYRETRYCOUNT 40

exttrail ./dirdat/et

--the default, causes Extract to write only the primary key and the

--changed columns of a row to the trail for update operations. This provides enough

--information to update the correct target record, while restricting the amount of data
that must be processed

COMPRESSUPDATES

GETTRUNCATES

WILDCARDRESOLVE DYNAMIC

table ggstrn01.*;

GGSCI (GGSEDU50) 22> view param dump_mss

extract dump_mss

passthru

rmthost 192.168.40.128, mgrport 7809

rmttrail c:/gg-sql/dirtrail/rt
```

```
gettruncates

wildcardresolve dynamic

table ggstrn01.*;

GGSCI (GGSEDU50) 24> start extract ext_mss

Sending START request to MANAGER ('GGMGRSQLSERVER') ...

EXTRACT EXT_MSS starting

GGSCI (GGSEDU50) 32> info exttrail ./dirdat/et, detail

Extract Trail: ./dirdat/et

Extract: EXT_MSS

Seqno: 0

RBA: 804

File Size: 50M

GGSCI (GGSEDU50) 34> start extract DUMP_MSS

Sending START request to MANAGER ('GGMGRSQLSERVER') ...

EXTRACT DUMP_MSS starting

GGSCI (GGSEDU50) 49> info all

Program    Status    Group      Lag        Time Since Chkpt

MANAGER    RUNNING

EXTRACT    RUNNING   DUMP_MSS   00:00:00   01:37:37

EXTRACT    RUNNING   EXT_MSS    00:00:00   00:00:02
```


REPLICAT	STOPPED	REP_GGS	00:00:00	1518:33:06
REPLICAT	STOPPED	RORA_SY	00:00:00	1518:33:05

Extract 参数说明：

Extract :指定提取进程的名称

NOCATALOGCONNECT：默认,提取进程 Extract 和复制进程 Replicat 创建一个访问目录的新连接

FETCHBATCHSIZE：初始化数据加载时，启用批量提取改善性能，而不是一次只去一行.如果提取的是 LOB 数据，还是每次提取一行。

SOURCEDB：就是指定源数据库的名称、用户名、密码

GETTRUNCATES：把 TRUNCATE 动作看作 DML 来进行处理

wildcardresolve dynamic：表示不先读取数据字典的信息，等 Extract 提取数据时，再读取数据字典信息，这样就可以指定表的时候，直接写 schema.*

COMPRESSUPDATES：默认值, 提取进程 Extract 只写 primary key 和行里变化列的数据到 trail 文件，这已经提供了足够用于做 update 操作的信息，而不是把所有列的信息都写到 trail 文件。

exttrail：表明提取进程提取日志以后，把数据写到本地文件的位置

详细参数说明参考：

http://download.oracle.com/docs/cd/E15881_01/doc.104/gg_wux_ref_v104.pdf

DUMP 参数说明：

Passthru：因为 DUMP 就是路由的功能，做数据网络传输，这个参数就是表示数据不做任何转换就写到远程 trail 文件。

rmthost：指定远程数据库服务器的 IP 地址和管理端口（默认 7809）

rmtrail：指定 DUMP 把本地 trail 文件写到远程什么位置。

进程参数设置好以后，通过 start 命令启动所有进程，然后通过 info all 进行验证，如果所有都是 running 的状态，就表明进程已经识别数据库信息，当有任何事务提交时，就会读取日志并且写 trail 文件。

- **源数据库提取进程配置好以后，我们要在目标数据库配置传递进程 replicat**

```
GGSCI (GGSEDU50) 70> ADD REPLICAT REP_MSS, EXTTRAIL c:/gg-sql/dirtrail/rt,  
checkpointtable ggstrn02.checkpoint  
  
REPLICAT added.
```

```
GGSCI (GGSEDU50) 58> view param rep_mss

replicat rep_mss

--dboptions nolimitrows

--showsyntax

batchsql

targetdb db02, userid ggstrn02, password ggstrn02

assumetargetdefs

--handlecollisions

discardfile ./dirrpt/rep_mss.dsc, append

repererror default, discard

gettruncates

wildcardresolve dynamic

map ggstrn01.*, target ggstrn02.*;
```

参数说明：

replicat ：指定复制进程的名称

dboptions nolimitrows ：如果没有主键、唯一键，做 DML 操作时会一行一行处理，如果希望批量处理用这个参数

showsyntax ：指定复制时候是否显示应用的 SQL 语句

batchsql ：如果有批量的 DML，复制进程可以把它们作为一个事务一次性

提交

targetdb : 指定目标数据库的名称、用户名、密码

assumetargetdefs : 表示源端数据库和目标端数据库的表结构一致

--handlecollisions : 如果出现唯一性约束的冲突, 自动解决冲突

discardfile : 如果出现不符合规则的数据, 放到这个参数指定的文件, 为后期处理

期处理

Gettruncates : 把 truncate 当作 DML 语句处理

map ggstrn01.*, target ggstrn02.* : 表明从源库那个用户下的表, 复制到

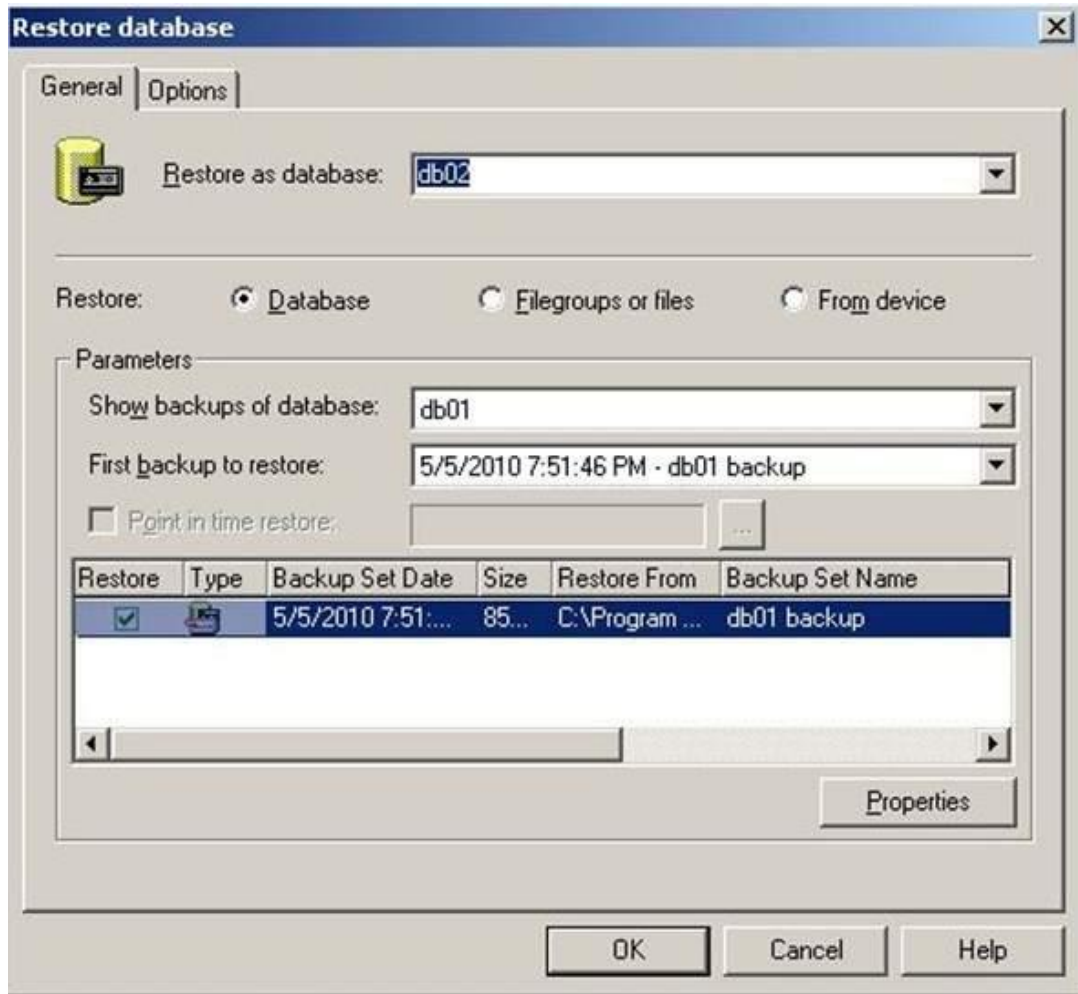
目标数据库下哪个用户的表

- **在目标数据库还原我们之前在源库 DB01 做的全库备份集**

1、all tasks->take offline db02

2、select options-> select "force restore over existing database"

3、click ok



- 因为数据库是从源库还原过来的，源库的数据库用户是 **ggstrn01**，要改为目标数据库的用户 **ggstrn02**

```
DECLARE @T_Name VARCHAR(100),@T_Owner VARCHAR(50),@SQL
VARCHAR(500),@T_Old VARCHAR(50),@T_New VARCHAR(50)

SET @T_Old='ggstrn01'

SET @T_New='ggstrn02'

DECLARE t_owner CURSOR FOR

SELECT [NAME],User_NAME(uid) AS UserName FROM SYSOBJECTS WHERE XTYPE<>'S'
```



```
OPEN t_owner

FETCH NEXT FROM t_owner INTO @T_Name,@T_Owner

WHILE @@FETCH_STATUS=0

BEGIN

    IF UPPER(@T_Owner)=UPPER(@T_Old)

        BEGIN

            SET @SQL=' sp_changeobjectowner '''+@T_Old+'.'+@T_Name+'',''+@T_New+''''

            --PRINT @SQL

            EXEC(@SQL)

        END

    FETCH NEXT FROM t_owner INTO @T_Name,@T_Owner

END

CLOSE t_owner

DEALLOCATE t_owner
```

- **指定从备份点的 LSN 开始复制，在源库查找 LSN**

```
SELECT bf1.physical_device_name, bs1.position, bs1.type,

bs1.last_lsn, backup_start_date, bs1.backup_finish_date

FROM msdb..backupset bs1

inner join

msdb..backupmediafamily bf1
```

```

inner join

msdb..backupmediafamily bf1

on bf1.media_set_id = bs1.media_set_id

WHERE bs1.database_name = 'db01'

AND bs1.type = 'D'

ORDER BY bs1.backup_start_date DESC

physical_device_name          position  type last_lsn
backup_start_date      backup_finish_date

-----

C:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\db01backup    1      D
5000000039000001          2010-05-05 19:51:46.000    2010-05-05 19:51:46.000

(1 row(s) affected)

select count(*) from db01.ggstrn01.emp;

-----

101

(1 row(s) affected)

select count(*) from db02.ggstrn02.emp;

-----

1

(1 row(s) affected)

```

因为是从源库备份还原过来的数据，表明源库在备份后又做了 100 条数据的

插入。

- **通过提取进程 Extract , 让提取进程从备份后的 LSN 开始提取数据**

```
GGSCI (GGSEDU50) 4> stop extract ext_mss

Sending STOP request to EXTRACT EXT_MSS ...

Request processed.

GGSCI (GGSEDU50) 6> alter extract ext_mss, lsn 5000000039000001

EXTRACT altered.

GGSCI (GGSEDU50) 7> start extract ext_mss

Sending START request to MANAGER ('GGMGRSQLSERVER') ...

GGSCI (GGSEDU50) 9> info ext_mss, detail

EXTRACT  EXT_MSS  Last Started 2010-05-06 02:38  Status RUNNING

Checkpoint Lag    00:00:00 (updated 00:00:01 ago)

VAM Read Checkpoint 2010-05-05 22:34:50.450000 LSN: 0x00000005:0000018b:0069

Target Extract Trails:

Remote Trail Name          Seqno    RBA    Max MB
./dirdat/et                1    11613    50

Extract Source             Begin      End
Not Available              * Initialized * 2010-05-05 22:34
Not Available              * Initialized * First Record
```

```
Not Available                2010-05-06 00:11  2010-05-06 00:11

Not Available                * Initialized *  2010-05-06 00:11

Current directory   C:\gg-sql

Report file        C:\gg-sql\dirrpt\EXT_MSS.rpt

Parameter file     C:\gg-sql\dirprm\EXT_MSS.prm

Checkpoint file    C:\gg-sql\dirchk\EXT_MSS.cpe

Process file       C:\gg-sql\dirpcs\EXT_MSS.pce

Error log          C:\gg-sql\ggserr.log

GGSCI (GGSEDU50) 10> sh logdump

Oracle GoldenGate Log File Dump Utility

Version 10.4.0.19 Build 002

Copyright (C) 1995, 2009, Oracle and/or its affiliates. All rights reserved.

Logdump 552 >open ./dirdat/et000000

Current LogTrail is C:\gg-sql\dirdat\et000000

Logdump 553 >count

LogTrail C:\gg-sql\dirdat\et000000 has 1 records

Total Data Bytes      796

Avg Bytes/Record      796

Others                1

Average of 1 Transactions

Bytes/Trans .....    844
```

```
Records/Trans ...      1

Files/Trans .....      1

Logdump 554 >nexttrail

LogTrail C:\gg-sql\dir\et000000 closed

Current LogTrail is C:\gg-sql\dir\et000001

Logdump 555 >count

LogTrail C:\gg-sql\dir\et000001 has 102 records

Total Data Bytes      5196

Avg Bytes/Record      50

Insert                100

RestartOK              1

Others                 1

After Images          101

Average of 3 Transactions

Bytes/Trans .....    3364

Records/Trans ...     34

Files/Trans .....      1
```

通过 alter extract 指定从特定的 LSN 开始提取日志信息，通过 logdump 工具，我们检查提取的 trail 文件，利用 count 命令，我们知道已经提取了所有的事务。

- 在目标数据库创建一个 checkpoint table

```
GGSCI (GGSEDU50) 5> dblogin sourcedb db02, userid ggstrn02, password ggstrn02

Successfully logged into database.

GGSCI (GGSEDU50) 6> add checkpointtable ggstrn02.checkpoint

Successfully created checkpoint table GGSTRN02.CHECKPOINT.
```

Checkpoint table 可以记录事务开始的 LSN、事务提交的 LSN 以及事务失败时，replicat 进程知道从哪个 LSN 开始恢复。

- 启动复制进程

```
GGSCI (GGSEDU50) 74> start replicat rep_mss, aftercsn 5000000039000001

Sending START request to MANAGER ('GGMGRSQLSERVER') ...

REPLICAT REP_MSS starting

GGSCI (GGSEDU50) 75> stats rep_mss

Sending STATS request to REPLICAT REP_MSS ...

Start of Statistics at 2010-05-06 04:13:53.

Replicating from GGSTRN01.EMP to GGSTRN02.EMP:

*** Total statistics since 2010-05-06 04:13:51 ***

Total inserts          100.00

Total updates          0.00
```


Total deletes	0.00
Total discards	0.00
Total operations	100.00

*** Daily statistics since 2010-05-06 04:13:51 ***

Total inserts	100.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	100.00

*** Hourly statistics since 2010-05-06 04:13:51 ***

Total inserts	100.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	100.00

*** Latest statistics since 2010-05-06 04:13:51 ***

Total inserts	100.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	100.00

End of Statistics.

GGSCI (GGSEDU50) 77> info replicat rep_mss, detail

REPLICAT REP_MSS Last Started 2010-05-06 04:13 Status RUNNING

Checkpoint Lag 00:00:00 (updated 00:00:07 ago)

Log Read Checkpoint File c:/gg-sql/dirtrail/rt000000

2010-05-05 22:34:50.450000 RBA 11660

Extract Source	Begin	End
c:/gg-sql/dirtrail/rt000000	* Initialized *	2010-05-05 22:34
c:/gg-sql/dirtrail/rt000000	* Initialized *	First Record

Current directory C:\gg-sql

Report file C:\gg-sql\dirrpt\REP_MSS.rpt

Parameter file C:\gg-sql\dirprm\REP_MSS.prm

Checkpoint file C:\gg-sql\dirchk\REP_MSS.cpr

Checkpoint table GGSTRN02.CHECKPOINT

Process file C:\gg-sql\dirpcs\REP_MSS.pcr

Error log C:\gg-sql\ggserr.log

我们的编辑团队

您若有何意见与建议，欢迎[与我们的编辑联系](#)。

诚挚感谢以下人员热情参与 TechTarget 中国《Oracle 系列电子书》的内容编辑工作！

诚邀更多的数据库专业人士加入我们的内容建设团队！



赵宇

TechTarget 中国特邀技术专家。05年开始研究Oracle, 从事过电信级的数据库开发和管理工作。目前专注于ODI、GoldenGate等新产品技术，长于为客户做性能优化和问题诊断。