sqlldr学习
http://docs.oracle.com/cd/E11882_01/server.112/e22490/ldr_concepts.htm#SUTIL003

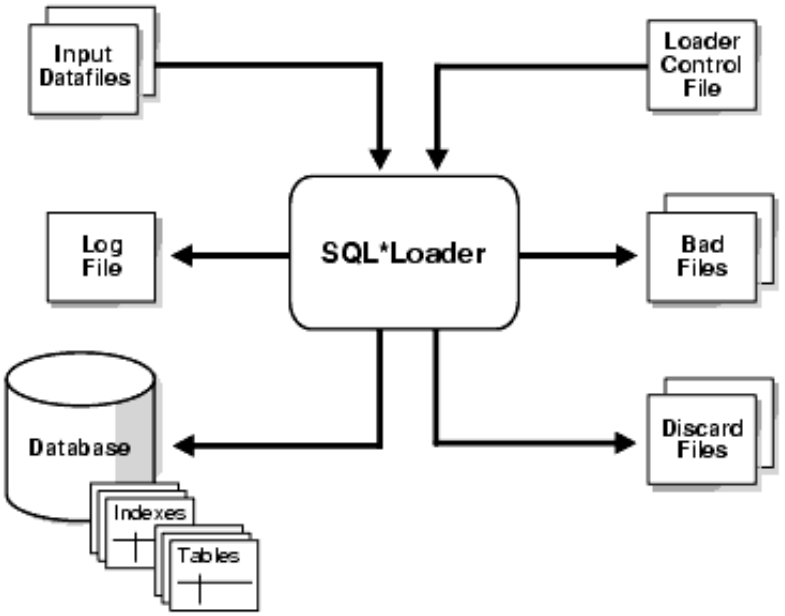
简介:
是一个对数据装载入库的工具,适用于文本数据
常见类似工具:
imp:只适合exp工具导出的文件
impdp:只适合expdp工具导出的文件
外部表:适合文本数据,但是只能本地机器,不能远程装载
sqlldr:适合文本数据,使用简单,相对外部表更加灵活,所以对文本装载时多数采用SQL Loader

可以实现不同数据库的数据交换

特性：
1. 通过网络来加载数据，意味着能够在一个不同的系统上运行sqlldr来加载数据
2. 在一个load会话内，导入多个datafile
3. 在一个会话内，数据能够加载到多个表
4. 指定数据的字符集
5. 选择性的加载数据,when和skip,
6. 能够在load进去之前使用SQL函数,系统的,也可以是自己写的函数,
7. 生成唯一键值（可做主键）
8. 使用OS的文件系统访问数据文件,例如在Windows上将NTFS数据文件传输至
9. 数据能够从磁盘，磁带，命名管道加载
10. 生成详细的错误报告，方便故障排除
11. 能够load任意复杂的对象关系数据（主外键）
12. 使用辅助数据文件来完成加载lob和集合
13. 能够使用传统load和直接load，传统的灵活，直接的性能优越

注意:只能加载到现有表,不能创建表
权限要求:
使用insert或者append,需要insert权限
使用replace或者truncate,需要delete权限,replace就是先delete原表中的所有表,再插入

常用构成方式:
控制文件:里面包含怎样去处理数据文件的信息
数据文件:按照某种格式进行排列的数据的集合




```
H:\>sqlldr

SQL*Loader: Release 11.2.0.4.0 - Production on 星期二 2月 21 15:41:07 2017

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.


用法: SQLLDR keyword=value [,keyword=value,...]

有效的关键字:

    userid -- ORACLE 用户名/口令
   control -- 控制文件名
       log -- 日志文件名
       bad -- 错误文件名
      data -- 数据文件名
   discard -- 废弃文件名
discardmax -- 允许废弃的文件的数目         (全部默认)
      skip -- 要跳过的逻辑记录的数目  (默认 0)
      load -- 要加载的逻辑记录的数目  (全部默认)
    errors -- 允许的错误的数目          (默认 50)
      rows -- 常规路径绑定数组中或直接路径保存数据间的行数
                (默认: 常规路径 64, 直接路径全部)
  bindsize -- 常规路径绑定数组的大小 (以字节计)  (默认 256000)
    silent -- 运行过程中隐藏消息 (标题,反馈,错误,废弃,分区)
    direct -- 使用直接路径                      (默认 FALSE)
   parfile -- 参数文件: 包含参数说明的文件的名称
  parallel -- 执行并行加载                    (默认 FALSE)
      file -- 要从以下对象中分配区的文件
skip_unusable_indexes -- 不允许/允许使用无用的索引或索引分区  (默认 FALSE)
skip_index_maintenance -- 没有维护索引, 将受到影响的索引标记为无用  (默认 FALSE)
commit_discontinued -- 提交加载中断时已加载的行  (默认 FALSE)
  readsize -- 读取缓冲区的大小              (默认 1048576)
```

```
external_table -- 使用外部表进行加载; NOT_USED, GENERATE_ONLY, EXECUTE  (默认 NOT_USE
columnarrayrows -- 直接路径列数组的行数  (默认 5000)
streamsize -- 直接路径流缓冲区的大小 (以字节计)  (默认 256000)
multithreading -- 在直接路径中使用多线程
 resumable -- 对当前会话启用或禁用可恢复  (默认 FALSE)
resumable_name -- 有助于标识可恢复语句的文本字符串
resumable_timeout -- RESUMABLE 的等待时间 (以秒计)  (默认 7200)
date_cache -- 日期转换高速缓存的大小 (以条目计)  (默认 1000)
no_index_errors -- 出现任何索引错误时中止加载  (默认 FALSE)

PLEASE NOTE: 命令行参数可以由位置或关键字指定
。前者的例子是 'sqlldr
scott/tiger foo'; 后一种情况的一个示例是 'sqlldr control=foo
userid=scott/tiger'。位置指定参数的时间必须早于
但不可迟于由关键字指定的参数。例如,
允许 'sqlldr scott/tiger control=foo logfile=log', 但是
不允许 'sqlldr scott/tiger control=foo log', 即使
参数 'log' 的位置正确。
```

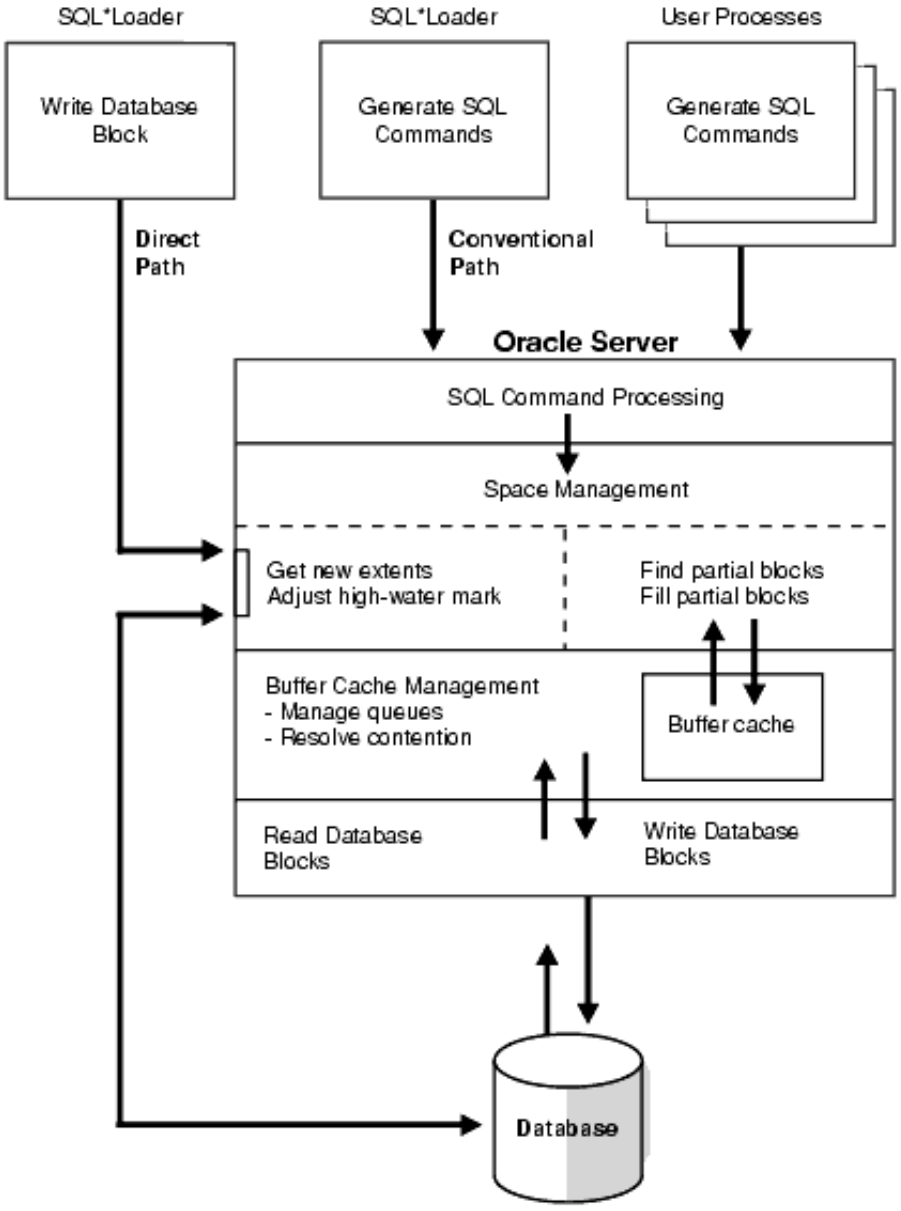
需要解释的参数:
BAD:指定文件名,默认为<数据文件名>.bad, 存放不满足约束的值
ERROR:默认50,允许错误的数目,此处错误指bad,达到数量后终止load过程,并且提交该点之前的所有数据
DISCARD:指定文件名,默认为<数据文件名>.dsc, 不满足条件而被丢弃的值,当不满足控制文件中的任意一个when,或者所有字段都为NULL,那么会被放入此文件中
DISCARDMAX:默认不限制,当达到丢弃限制后,数据文件的处理终止并且继续下一个文件

rows:常规路径中,进行插入时绑定的数据的行数
bindsize:可以看做是rows占用的缓存,并且当缓存设置不够时,rows会受bindsize限制

date_cahce:
默认值：Enabled（对于1000个以内的不同日期）。要完全禁用日期缓存功能，请将其设置为0。仅适用于直接路径加载
日期缓存用于存储从文本字符串到内部日期格式的转换结果。缓存很有用，因为查找文本对应的日期的成本远远少于从文本格式到日期格式的转换。如果在数据文件中重复出现相同的日期，则使用日期缓存可以提高直接路径加载的速度。
DATE_CACHE指定日期缓存大小（在entries中）。例如，DATE_CACHE=5000指定每个创建的日期缓存最多可以包含5000个不同的日期条目。每个表都有自己的日期缓存，如果需要的话。仅当至少一个日期或时间戳值被加载，并需要进行数据类型转换才能存储在表中时，才创建日期高速缓存。
日期缓存功能仅适用于直接路径加载。默认情况下启用。默认日期缓存大小为1000个元素。如果使用默认大小，并且加载的唯一输入值数量超过1000，则会自动为该表禁用日期缓存功能。但是，如果覆盖默认值并指定非零的日期缓存大小，并且超过了该大小，则不会禁用缓存。关于是否自动替换这个事,并没有想到什么对策来验证
您可以使用日志文件中包含的日期高速缓存统计信息（entries, hits, and misses）来调整高速缓存

的大小，以便将来进行类似加载。

DIRECT:直接加载
默认false,使用常规方式
当改之为true时,启动直接路径方式



常规方式:
就是使用普通的insert语句,sql*loader用户进程构建values子句中包含绑定变量的insert语句,然后读取源文件为每个要插入的行执行一次insert,使用buffercache,生成redo和undo,通过普通的提交处理实现数据永久化.

直接路径:
避开数据库缓冲区缓存,SQL loader读取源文件,并将内容发送到服务器进程,然后服务器进程直接在其PGA中组装表数据中的块,并且将他们直接写入到数据文件,写操作在表的高水位线上完成,加载完成之后,SQLloader会移动高水位线,使其包含最新写入的数据,并且其他用户都能立刻看见这些数据块的行,不会生成撤销,因为和SGA的交互最小,所以不会对最终用户产生影响

直接路径加载极快,但是存在下述缺点:
1．执行操作期间,必须删除或者禁用引用完整性约束(就是主外键的关系),注意做实验时发现也会违反唯一性约束
2．没有激活insert触发器
3．将针对其他会话的DML锁定表
4．无法为群集表使用直接路径

MULTITHREADING 多线程,
默认值:在多CPU系统上TRUE,在单CPU上默认为FALSE
仅适用于直接路径加载

NO_INDEX_ERRORS
默认值:none
仅对直接路径加载有效
当NO_INDEX_ERRORS在命令行上指定时，索引不会在加载期间的任何时间设置为不可用。如果检测到任何索引错误，则中止装载。也就是说，不加载任何行，并且索引保持不变

PARALLEL
默认值 FALSE
指定直接装入是否可在多个并行会话中操作以将数据装载到同一个表中

PARFILE
能够指定包含常用命令行参数的文件的名称,例如

```
USERID=scott
CONTROL=daily_report.ctl
ERRORS=9999
LOG=daily_report.log
```

但是这些参数大部分是能够在控制文件中指定的,功能不大

READSIZE:读缓冲区大小
参数默认值可以使用不带任何参数的sqlldr看到
READSIZE是负责读取的缓冲区大小,
注意和bindsize区分,bindsize是负责提交的缓冲区大小
仅适用于从数据文件中读取数据使用,对于数据放在控制文件中的不适用(此情况默认64KB)
注意此值的最大值是和OS平台相关的
此值在传统模式下可由bindsize覆盖,也就是说,假如此值比bindsize小,那么该值会增大

RESUMABLE:
默认false

该RESUMABLE参数用于启用和禁用可恢复空间分配。由于默认情况下禁用此参数，因此必须设置RESUMABLE=true为使用其关联的参数RESUMABLE_NAME和RESUMABLE_TIMEOUT。
参照ORACLE的RESUMABLE特性,当正在load然后发现空间不足的时候,等待RESUMABLE_TIMEOUT的时间,DBA抓紧扩容表空间或存储之后,能够继续执行,否则中断并返回异常
此功能是为了以防万一

RESUMABLE_NAME
默认： 'User USERNAME (USERID), Session SESSIONID, Instance INSTANCEID'
此参数的值标识可恢复的语句。此值是插入在USER_RESUMABLE或DBA_RESUMABLE视图中的用户定义的文本字符串，以帮助您标识已暂停的特定恢复语句。
除非RESUMABLE将此参数设置true为启用可恢复空间分配，否则将忽略此参数。

RESUMABLE_TIMEOUT
默认值：7200秒（2小时）
参数的值指定必须固定错误的时间段。如果错误在超时期限内不固定，则语句的执行终止，而不完成。
除非RESUMABLE将此参数设置true为启用可恢复空间分配，否则将忽略此参数。

ROWS (rows per commit):
这个是每次加载的行数
注意和上面的bindsize,readsize相区分
rows:每次提交的记录数

- bindsize:每次提交记录的缓冲区
- readsize:每次读取的文件的缓冲区,注意碰到bindsize不同时,较小者自动调整到较大者
- rows和bindsize会先计算单条的记录长度,再乘以rows,不会试图扩展rows以填充bindsize,假如超过了那么以bindsize为准(使用较小值)

这三个参数是相互补充的

- 仅传统路径加载：此ROWS参数指定绑定数组中的行数。最大行数为65534.请参见“绑定数组和常规路径加载”。
- 仅限直接路径装入：此ROWS参数标识在数据保存之前要从数据文件读取的行数。默认值是在加载结束时读取所有行并保存数据一次。加载到保存表中的实际行数大约为ROWS减去自上次保存以来丢弃和被拒绝的记录数。

SILENT 调整反馈模式
您可以通过SILENT使用一个或多个值指定来抑制这些消息。

例如，您可以使用以下命令行参数抑制屏幕上通常显示的标题和反馈消息：

SILENT =（HEADER，FEEDBACK）
使用适当的值抑制以下一个或多个：

- HEADER - 抑制通常出现在屏幕上的SQL * Loader标题消息。标题消息仍显示在日志文件中。
- FEEDBACK - 抑制通常出现在屏幕上的“达到提交点”反馈消息。
- ERRORS - 抑制记录生成导致将其写入到坏文件的Oracle错误时出现的日志文件中的数据错误消息。仍显示已拒绝的记录数。
- DISCARDS - 针对写入丢弃文件的每条记录，抑制日志文件中的消息。
- PARTITIONS - 在直接加载分区表期间，禁止将每个分区的统计信息写入日志文件。
- ALL- 实现所有的抑制值：HEADER, FEEDBACK, ERRORS, DISCARDS,和PARTITIONS.

SKIP :要跳过的记录
默认值:不跳过任何记录
SKIP 指定不应加载的文件开头的逻辑记录数。
此参数用于继续由于某种原因中断的负载,当将相同数量的记录load到每个表中时,它用于所有常规load，单表直接load和多表直接load,当将不同数量的记录加载到每个表中时，不用于多表直接load。如果WHEN还存在子句并且load涉及辅助数据，则只有在WHEN子数据文件中的记录成功时，辅助数据才会被跳过。

SKIP_INDEX_MAINTENANCE
默认false
该SKIP_INDEX_MAINTENANCE参数停止直接路径load的索引维护
仅适用于直接路径加载
它导致将已添加了索引键的索引分区标记为“索引不可用”，因为索引段与其索引的数据不一致。在加载前就不可用的索引不受影响
该SKIP_INDEX_MAINTENANCE参数：

- 适用于本地和全局索引
- 可以使用（与PARALLEL参数）对具有索引的对象执行并行加载
- 可以使用（与PARTITION该参数INTO TABLE子句）做单个分区负载到具有全局索引的表
- 将索引和索引分区的列表（在SQL * Loader日志文件中）设置为索引不可用状态

SKIP_UNUSABLE_INDEXES
默认值：Oracle数据库配置参数的值SKIP_UNUSABLE_INDEXES，如初始化参数文件中指定。默认数据库设置为TRUE。

SQL*Loader和Oracle数据库都提供了一个SKIP_UNUSABLE_INDEXES参数。SQL*Loader SKIP_UNUSABLE_INDEXES参数在SQL*Loader命令行中指定。Oracle数据库SKIP_UNUSABLE_INDEXES参数在初始化参数文件中指定为配置参数。重要的是要了解他们如何互相影响。

如果SKIP_UNUSABLE_INDEXES在SQL * Loader命令行中指定值，则它将覆盖SKIP_UNUSABLE_INDEXES初始化参数文件中的配置参数的值。

如果不在SKIP_UNUSABLE_INDEXESSQL * Loader命令行中指定值，则SQL * Loader将使用

SKIP_UNUSABLE_INDEXES初始化参数文件中指定的配置参数的数据库设置。如果初始化参数文件未指定数据库设置SKIP_UNUSABLE_INDEXES，则默认数据库设置为TRUE。

SKIP_UNUSABLE_INDEXES值为TRUE表示如果遇到索引不可用状态下的索引，则跳过该索引，并继续加载操作。这允许SQL*Loader加载具有在加载开始之前处于不可用状态的索引的表。在加载可用状态的索引将由SQL*Loader维护。在加载时处于不可用状态的索引将不被维护，在加载完成时将保持在不可用状态。

STREAMSIZE
默认值,调用sqlldr查看
streamsize -- 直接路径流缓冲区的大小 (以字节计) (默认 256000)
指定直接路径流的大小（以字节为单位）。

结果	退出代码
所有行已成功加载	EX_SUCC
所有或部分行被拒绝	EX_WARN
全部或部分行被丢弃	EX_WARN
停止加载	EX_WARN
命令行或语法错误	EX_FAIL
Oracle *不可恢复的SQL * Loader的错误	EX_FAIL
操作系统错误（如文件打开/关闭和malloc）	EX_FAIL

对于UNIX，退出代码如下：

- EX_SUCC 0
- EX_FAIL 1
- EX_WARN 2
- EX_FTL 3

对于Windows NT，退出代码如下：

- EX_SUCC 0
- EX_FAIL 1
- EX_WARN 2
- EX_FTL 4

```
1    -- This is a sample control file
2    LOAD DATA
3    INFILE 'sample.dat'
4    BADFILE 'sample.bad'
5    DISCARDFILE 'sample.dsc'
6    APPEND
7    INTO TABLE emp
8    WHEN (57) = '.'
9    TRAILING NULLCOLS
10  (hiredate SYSDATE,
      deptno POSITION(1:2)  INTEGER EXTERNAL(2)
             NULLIF deptno=BLANKS,
       job    POSITION(7:14)  CHAR  TERMINATED BY WHITESPACE
             NULLIF job=BLANKS  "UPPER(:job)",
       mgr    POSITION(28:31) INTEGER EXTERNAL
             TERMINATED BY WHITESPACE, NULLIF mgr=BLANKS,
       ename  POSITION(34:41) CHAR
             TERMINATED BY WHITESPACE  "UPPER(:ename)",
       empno  POSITION(45) INTEGER EXTERNAL
             TERMINATED BY WHITESPACE,
       sal    POSITION(51) CHAR  TERMINATED BY WHITESPACE
             "TO_NUMBER(:sal,'$99,999.99')",
       comm   INTEGER EXTERNAL  ENCLOSED BY '(' AND '%'
             ":comm * 100"
    )
```

```
1.  LOAD DATA
2.  INFILE *
3.  INTO TABLE TT
4.  REPLACE
5.  FIELDS TERMINATED BY ','
6.  (
7.  ID,
8.  COMMENTS "REPLACE(:COMMENTS,'/n',CHR(10))"
9.  )
```

这个列能够捕捉换行,并将换行replace成为oracle中的换行符CHR(10)

在导入大量数据的时候i，sqlldr往往不敬人意，在导入的时候存在效率问题，可以通过一下几种方式提

高sqlldr的速度.

1. 使用direct=Y 这是速度提高最快的方式
2. 使用大的readsize/streamsize 提高读写数据的缓冲区的大小
3. 使用大的bindsize (只对conventional path load有效) , 提高一次提交的数据量, 效果也比较明显.
4. 使用大的columnarrayrows
Number of rows for direct path column array 提高direct load的效率,
5. 使用并行load, 最好运行两个不同的sqlldr程序, 指定每个ctl文件加载不同的内容.使用skip
6. 先删除加载表上的索引和约束. 具体效果不是很清楚, 但是, 肯定可以提高加载的速度.
7. 最好将加载表设置成nologging/unrecoverable, 对于conventional path load比较有用, 减少重做日志的写入.
8. 加大date_cache的大小, 在加载表有日期字段的时候能够提高加载的速度.
date_cache -- size (in entries) of date conversion cache (Default 1000)

```
01,Vivian,newyork
02,Anna,
03,Cona,shanghai
04,,beijing
10,dongsc,jinan
```

```
INTO TABLE dept
   WHEN recid = 1
   (recid  FILLER POSITION(1:1)  INTEGER EXTERNAL,
    deptno POSITION(3:4)  INTEGER EXTERNAL,
    dname  POSITION(8:21) CHAR)
INTO TABLE emp
   WHEN recid <> 1
   (recid  FILLER POSITION(1:1)   INTEGER EXTERNAL,
    empno  POSITION(3:6)   INTEGER EXTERNAL,
    ename  POSITION(8:17)  CHAR,
    deptno POSITION(19:20) INTEGER EXTERNAL)
```

如果数据文件的字符集和数据库的字符集不一样，SQL*Loader会自动把数据文件的字符集转换成数据库的字符集，当然前提条件是数据库的字符集是数据文件的字符集的超集。

数据文件的字符集可以通过控制文件中的CHARACTERSET参数配置，其语法如下：
CHARACTERSET char_set_name
如果没有设置CHARACTERSET参数，数据文件的字符集由操作系统的NLS_LANG设置。

还有一种字符集要特别注意，就是控制文件本身的字符集（只能由NLS_LANG设置），如果控制文件的字符集和数据文件的不一样，会先转换成数据文件的字符集，但这样很容易出错（特别是分隔符和非英文列名），因此，实际使用中为了方便，一般把NLS_LANG，CHARACTERSET（如果有的话）设成和数据库字符集一样。

```
load data
infile 't1.txt' badfile 't1.bad' discardfile 't1.dsc'
into table t1
append
WHEN id="10"
fields terminated by ','
trailing nullcols
(
id,
name,
addr
)
```

参数:
load data infile "users_data.csv" infile "users_data2.csv" ..... 导入的数据文件,可以使用infile指定多个
skip=n 跳过数据文件的n行,注意控制文件中假如有多个数据文件的话,那么只会跳过第一个数据文件的多少行

控制文件是SQL*Loader里最重要的文件，它是一个文本文件，用来定义数据文件的位置、数据的格式、以及配置数据加载过程的行为，在sqlldr中以control参数指定控制文件。
控制文件可以包含在命令行中的参数,并且控制文件还能包含数据,但是几乎没人这么做
有多个数据文件的情况:

```
INFILE  mydat1.dat  BADFILE  mydat1.bad  DISCARDFILE mydat1.dis
INFILE  mydat2.dat
INFILE  mydat3.dat  DISCARDFILE  mydat3.dis
INFILE  mydat4.dat  DISCARDMAX  10 0
```

存放的数据格式必须完全相同

```
1.
2.  OPTIONS (DIRECT=true,SKIP_INDEX_MAINTENANCE=true,PARALLEL=true)
3.  LOAD DATA
4.  INFILE 'nor.dat'
5.  BADFILE 'nor.bad'
6.  DISCARDFILE 'nor.dsc'
7.  INTO TABLE p95169.DISEASE_EXPERT_RELATION
8.  APPEND
9.  WHEN len='3'
10. FIELDS TERMINATED BY WHITESPACE
11. (
12.   len FILLER POSITION(1) CHAR,
13.   DISEASEEXPERTUUID EXPRESSION "SYS_GUID()",
14.   EXPERTUUID CHAR,
15.   DISEASEUUID CHAR,
16.   DISEASESORTCODE EXPRESSION "NULL",
17.   DISEASENAME CHAR,
18.   CREATEDTIME EXPRESSION "TO_CHAR(sysdate,'yyyymmddhh24miss')",
19.   MODIFIEDTIME EXPRESSION "TO_CHAR(sysdate,'yyyymmddhh24miss')"
20. )
21. INTO TABLE p95169.DISEASE_EXPERT_RELATION
22. APPEND
23. WHEN len='2'
24. FIELDS TERMINATED BY WHITESPACE
25. (
26.   len FILLER POSITION(1) CHAR,
27.   DISEASEEXPERTUUID EXPRESSION "SYS_GUID()",
28.   EXPERTUUID CHAR,
29.   DISEASEUUID EXPRESSION "NULL",
30.   DISEASESORTCODE EXPRESSION "NULL",
31.   DISEASENAME CHAR,
32.   CREATEDTIME EXPRESSION "TO_CHAR(sysdate,'yyyymmddhh24miss')",
33.   MODIFIEDTIME EXPRESSION "TO_CHAR(sysdate,'yyyymmddhh24miss')"
34. )
35. INTO TABLE p95169.DISEASE_EXPERT_RELATION
36. APPEND
37. WHEN len='1'
38. FIELDS TERMINATED BY WHITESPACE
39. (
40.   len FILLER POSITION(1) CHAR,
41.   DISEASEEXPERTUUID EXPRESSION "SYS_GUID()",
42.   EXPERTUUID CHAR,
43.   DISEASEUUID EXPRESSION "NULL",
44.   DISEASESORTCODE EXPRESSION "NULL",
45.   DISEASENAME EXPRESSION "NULL",
46.   CREATEDTIME EXPRESSION "TO_CHAR(sysdate,'yyyymmddhh24miss')",
47.   MODIFIEDTIME EXPRESSION "TO_CHAR(sysdate,'yyyymmddhh24miss')"
48. )
```