



SQL*Loaderエクスプレス・モードによる Oracleデータベースのロード



Oracle Databaseユーティリティ

2021年8月 | バージョン2.00

Copyright © 2021, Oracle and/or its affiliates

本書の目的

本書では、Oracle Databaseに含まれるOracle SQL*Loaderエクスプレス・モードの機能と強化点について概要を説明します。

免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意したOracle Software License and Service Agreementの諸条件に従うものとします。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料にするものでもありません。本書に記載されている機能の開発、リリースおよび時期については、オラクルの裁量により決定されます。

製品アーキテクチャの性質上、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

目次

本書の目的	1
免責事項	1
はじめに	3
エクスプレス・モードを使用するメリット	3
CSVファイルの概要	3
エクスプレス・モードを使用したロード	4
エクスプレス・ロードの例	4
SQL*Loaderエクスプレス・モードのカスタマイズ	5
フィールド終端文字の変更	5
データファイル内のフィールド順序の指定	5
表名と一致しないデータファイル名の指定	5
囲み文字の使用	6
改行を含むフィールドでの囲み文字の使用	6
その他のカスタマイズ	7
エクスプレス・モードのログ・ファイルを使用した制御ファイルの作成	7
結論	8



はじめに

SQL*Loaderの制御ファイルは、作成するのが難しいことがあります。SQL*Loaderはさまざまなタイプのデータファイルをロードできる柔軟性を備えていますが、その柔軟性により、SQL*Loader制御ファイルの作成が複雑になる場合があります。データファイルに含まれるフィールドが多いほど、エラーが発生する可能性が高くなります。

エクスプレス・モードは、Oracle Database 12c以降のSQL*Loaderに含まれる機能で、CSVファイルのロードをより簡単で高速にします。データファイルの大半はカンマ区切り値（CSV）形式です。エクスプレス・モードを使用すると、ほとんどのCSVファイルのロードで制御ファイルの作成が不要になり、SQL*Loaderのコマンドラインにパラメータをいくつか指定するだけで、CSVファイルをロードできます。この技術概要では、エクスプレス・モードの使用を開始するための情報を提供します。

エクスプレス・モードを使用するメリット

SQL*Loaderのエクスプレス・モードでは、SQL*Loader制御ファイルの作成およびテストにかかる時間と労力を節約できます。ロード・プロセスは、1つのSQL*Loaderコマンドと数個のパラメータを指定するだけで開始できます。

SQL*Loaderエクスプレス・モードの自己最適化プロセスにより、可能な限り最速のメカニズムを使用してデータファイルをロードします。まず、外部表を使用して、APPENDヒントを付けたパラレル挿入を実行しようとします。パラレル挿入では、同時に複数のプロセスによるデータファイルのロードが可能であるため、ロードの所要時間を短縮できます。特定のロード処理に対して、エクスプレス・モードで外部表を使用できない場合は、次に高速なメカニズムであるダイレクト・パス・ロードが使用されます。挿入文にAPPENDヒントを指定すると、Oracleデータベースはダイレクト・パス・ロードを使用します。ダイレクト・パス・ロードでは、Oracleデータ・ブロックの形式を設定し、データベース・ファイルに直接データ・ブロックを書き込むことで、OracleデータベースのINSERTによるオーバーヘッドの大半を解消します。

CSVファイルの概要

エクスプレス・モードを使用してロードできるのは、CSVファイルのデータのみです。CSVファイルには次のような特徴があります。

CSVファイルはテキスト・ファイルで、表内の1行に相当するデータがファイル内の各レコードに含まれる

各レコード内のフィールド順序は、すべてのレコードで同じです。

各レコードの末尾が改行文字である

UNIX系オペレーティング・システムでは、改行はラインフィード文字です。Microsoft Windowsでは、改行はキャレッジ・リターンとラインフィード文字です。

レコード内の各フィールド値が特定の文字（必須ではないが通常はカンマ）で区切られる

必ずしもカンマを使用する必要はなく、コロンや縦棒などの文字も終了文字として使用できます。簡潔にするため、ここでは、カンマ以外の終了文字を使用するファイルもCSVファイルと呼びます。レコード内最後のフィールドの後の終端文字は付けても付けなくても構いません。

CSVファイルのフィールド値が二重引用符などの文字で囲まれている

囲み文字は通常、二重引用符文字ですが、別の文字でも構いません。値を囲むことで、フィールド値に改行文字やフィールド終了文字を含めることができます。

囲み文字の使用は必須または任意にできる

CSVファイルで囲み文字を使用する場合、すべてのフィールドで必須にすることも、任意で使用することもできます。囲み文字の使用を任意にする場合、通常は、改行文字や終了文字を含むフィールド値のみに囲み文字を使用します。

“区切り”構文を使用する場合、フィールドに囲み文字を含めることができる

囲み文字を使用するCSVファイルで、フィールド値にその囲み文字が含まれている場合、フィールド値の囲み文字を2回繰り返す必要があります。これは“区切り構文”と呼ばれ、フィールドの途中にある囲み文字を、フィールドの終了を示す囲み文字から区別するための方法です。

エクスプレス・モードを使用したロード

SQL*LoaderのTABLEパラメータにより、エクスプレス・モードが始まります。TABLEパラメータの値は、SQL*Loaderがロードする表の名前です。TABLEパラメータだけが指定されている場合、SQL*Loaderは以下を前提にしてデータをロードします。

1. 現在のディレクトリ内で、表と同じ名前と“.dat”拡張子を持つデータファイルを探す
(データファイル名の`大文字/小文字`が、TABLEパラメータで指定された表名の`大文字/小文字`と一致する)
2. データファイル内のフィールドの順序が、表内の列の順序に一致すると仮定する
3. フィールドがカンマで区切られており、囲み文字は使用されていないと仮定する

エクスプレス・ロードの例

次に示すのは、表、データファイル、表のロードに使用するSQL*Loaderコマンドの簡単な例です。

表は次のコマンドで作成されます。

```
CREATE TABLE EMP
(EMPNO    number(4) not null,
 ENAME    varchar2(10),
 HIREDATE date,
 DEPTNO   number(2));
```

表のロードに使用するデータファイルの名前は、表の名前empに基づいてemp.datとなっています。この例でのデータファイルの内容は次のとおりです。

```
7782,Clark,09-Jun-81,10
7839,King,17-Nov-81,12
```

データファイル内のフィールドの順序は、表内の列の順序と一致しています。次のSQL*Loaderコマンドにより、データファイルから表へのロードが実行されます。

```
sqlldr userid=scott table=emp
```

ここで制御ファイルは使用されていません。SQL*Loaderコマンドを実行した後、表にSELECTを実行すると次の結果が返されます。

EMPNO	ENAME	HIREDATE	DEPTNO
7782	Clark	09-JUN-81	10
7839	King	17-NOV-81	10

SQL*Loaderエクスプレス・モードのカスタマイズ

上の例のSQL*Loaderコマンドはごくシンプルなもの、デフォルト設定を使用していましたが、デフォルト設定が常に当てはまるとは限りません。たとえば、CSVファイルで、カンマ以外の終端文字が使用されることがあります。また、ロードに使用されるデータファイルが複数あり、ファイル名に表の名前が使用されないケースもあります。こういった理由から、SQL*Loaderエクスプレス・モードには追加のコマンドライン・パラメータがあり、ロード・プロセスをカスタマイズできるようになっています。

フィールド終端文字の変更

SQL*LoaderコマンドラインのTERMINATED_BYパラメータにより、フィールドの終了を示す文字を指定できます。上の例のデータファイルが、区切り文字としてカンマではなく縦棒を使用する場合、ファイルの内容は次のようになります。

```
7782|Clark|09-Jun-81|10
7839|King|17-Nov-81|12
```

このデータファイルのロードに使用するコマンドは次のとおりです。

```
sqlldr userid=scott table=emp terminated_by='|'
```

データファイル内のフィールド順序の指定

SQL*Loaderエクスプレス・モードでデータファイルのレコードからフィールドを読み取るとき、デフォルトでは、データファイル内のフィールド順が表内の列の順序と一致すると想定されます。ただし、これが当てはまらない場合もあります。その場合、FIELD_NAMESパラメータを使用できます。SQL*Loaderは、データファイル内の最初のレコードを使用してデータファイル内のフィールド順を決定します。

FIELD_NAMESパラメータには2つのオプションがあります。複数のデータファイルをロードする場合、FIELD_NAMES=FIRSTを指定すると、リスト内の1番目のデータファイルのみが最初のレコードにフィールド名リストを含むという意味になります。FIELD_NAMES=ALLを指定すると、すべてのデータファイルが最初のレコードにフィールド名リストを含むとSQL*Loaderは認識します。

次の例では、データファイルemp.datのフィールド順は、表EMPの列順と一致しません。データの1行目により、フィールドの順序が定義されます。

```
deptno,empno,ename,hiredate
10,7782,Clark,09-Jun-81
12,7839,King,17-Nov-81
```

このデータファイルのロードに使用するコマンドは次のとおりです。

```
sqlldr userid=scott table=emp field_names=first
```

表名と一致しないデータファイル名の指定

デフォルトでは、データファイルの名前は表名に.dat拡張子を付けたものです。別の名前を持つファイルをロードする場合や、複数ファイルをロードする場合、DATAパラメータを使用します。DATAパラメータには値のリストを指定できるので、複数のファイルをロードの対象にすることができます。

Oracle Database 12c以降のSQL*Loaderでは、ワイルドカード文字として"*"と"?"をファイル名に使用できます。ワイルドカード文字の"*"は、ファイル名に含まれる1文字または複数文字に相当します。ワイルドカード文字の"?"は、ファイル名に含まれる1文字のみに相当します。これにより、名前の類似した大量のファイルをロードする場合も、コマンド文字列をシンプルにできます。

たとえば、2つのファイルを表empにロードするとします。

最初のファイルjan_2012_emp.datの内容は次のとおりです。

```
7782,Clark,09-Jan-12,10
7839,King,17-Jan-12,12
```

2番目のファイルfeb_2012_emp.datの内容は次のとおりです。

```
8109,Baby,12-Feb-12,10
8299,Lee,24-Feb-12,12
```


これらのファイルをロードするには、次のコマンドのどちらでも使用できます。

```
sqlldr userid=scott table=emp data=jan_2012_emp.dat,feb_2012_emp.dat
sqlldr userid=scott table=emp data="*_2012_emp.dat"
```

最初のコマンドは、ロードするファイル名をすべて指定しています。2番目のコマンドでは、dataパラメータで指定するファイル名にワイルドカードを使用しています。

囲み文字の使用

データファイルのフィールドで、値の一部に終了文字や改行文字が含まれる場合、フィールドの前後に囲み文字が必要になります。これを指定するには、パラメータoptionally_enclosed_byを使用します。

たとえば、次の表定義について考えます。

```
create table part_info
    (part_number      varchar2(10),
     part_description varchar2(40)
    );
```

データファイルpart_info.datには、次のレコードが含まれています。

```
10-1002,"size 12 widget, red"
10-1003,"size 12 widget, blue"
```

表PART_INFOへのロードには、次のコマンドを使用できます。

```
sqlldr userid=scott table=part_info optionally_enclosed_by=\'\''
```

上の例でOPTIONALLY_ENCLOSED_BYを使用したのは、一部のフィールド値だけが引用符で囲まれていたからです。すべてのフィールドが囲まれている場合は、ENCLOSED_BYパラメータを使用します。

ロード後の表には、次のような行が含まれます。

PART_NUMBER	PART_DESCRIPTION
10-1002	size 12 widget, red
10-1003	size 12 widget, blue

改行を含むフィールドでの囲み文字の使用

囲み文字は、改行を含むフィールドにも使用できます。たとえば、大きいテキスト・フィールドを含む次の表定義があるとします。

```
CREATE TABLE EMP_RESUMES
    (EMPNO      number(4) not null,
     EMP_RESUME varchar2(1000));
```

データファイルには次に示す2つのレコード7782と7839が含まれています。7782,"line 1 of resume for 7782

```
line 2 of resume
last line of resume"
7839,"line 1 of resume for 7839
line 2 of resume
last line of resume"
```

EMP_RESUMEフィールドは二重引用符で囲まれており、データ内に改行が含まれています。この場合、SQL*Loaderコマンドで、CSVパラメータにWITH_EMBEDDEDという値を指定する必要があります。

注：ファイルのフィールド内に改行文字が含まれる場合、SQL*Loaderは追加のファイル処理を実行するため、ロード・プロセスのオーバーヘッドが増加し、処理時間が長くなります。

次のSQL*Loaderコマンドにより、上のデータファイルから表EMP_RESUMESへのロードが実行されます。

```
sqlldr userid=scott table=emp_resumes csv=with_embedded
```

表EMP_RESUMESに対してSELECTを実行すると、ファイルからロードされたデータが返されます。

```
SQL> select * from emp_resumes;
      EMPNO
```

```
-----
EMP_RESUME
```

```
-----
              7782
line 1 of resume for 7782
line 2 of resume
last line of resume
              7839
line 1 of resume for 7839
line 2 of resume
last line of resume
```

その他のカスタマイズ

SQL*Loaderエクスプレス・モードは、ほかにも、ロードをカスタマイズする以下のパラメータをサポートしています。詳細については、『Oracle Database ユーティリティ』のガイドを参照してください。

- BAD - ロードできなかったレコードをSQL*Loaderが書き込むファイル。
- CHARACTERSET - データファイルのエンコードに使用するキャラクタ・セットの名前。
- DATE_FORMAT - データファイル内の日付を解釈する際に使用するフォーマット文字列。
- DEGREE_OF_PARALLELISM - 外部表を使用してロードする際に使用する並列度。
- DIRECT - 外部表ではなくダイレクト・パスまたは従来型パス・ロードを使用するように指示します。
- EXTERNAL_TABLE = EXECUTEは、データファイルのロードに外部表を使用するように指示します。GENERATE_ONLYは、外部表のロードに使用するSQL文を、実行する代わりにログ・ファイルに書き込むよう指示します。このSQL文を編集およびカスタマイズし、SQL*Plusで実行することができます。
- LOAD - ロードするレコード数。デフォルトでは全レコードをロードします。
- NULLIF - レコード内のフィールド値がこの値である場合、SQL*LoaderはこのフィールドにNULLを挿入します。
- SILENT - SQL*Loaderのログ・ファイルに書き込むメッセージを制御します。
- TIMESTAMP_FORMAT - データファイルのタイムスタンプ・フィールドを解釈する際に使用するフォーマット文字列。
- TRIM - フィールドの最初と最後から空白をトリムする方法。

エクスプレス・モードのログ・ファイルを使用した制御ファイルの作成

SQL*Loaderのエクスプレス・モードで、複雑なSQL*Loaderの制御ファイルを生成することができます。これには、外部表を作成するためのDDLも含まれます。特定のファイルやファイル・セットをロードするのに、エクスプレス・モードのカスタマイズ・パラメータでは不十分な場合、この方法が役に立ちます。

SQL*Loaderのエキスプレス・モードを実行すると、通常のSQL*Loader制御ファイルの内容がログ・ファイルに書き込まれます。ログ・ファイルのデフォルト名は"tablename.log"です。制御ファイル情報をログ・ファイルに保存しておき、そのまま使用するか、カスタマイズのベースとして使用して、データをロードできます。

たとえば、データファイル内の日付フィールドごとにデータ形式が異なる場合、DATE_FORMATパラメータは使用できません。そのような場合は以下の手順を使用できます。

1. SQL*Loaderのエキスプレス・モードを使用して、制御ファイル情報をログ・ファイルに書き込む
2. この情報を使用して、新しい制御ファイルを作成する
3. 日付フィールドの正しい値に合わせて、データのフォーマット文字列を変更する

このように、SQL*Loaderのエキスプレス・モードは、より複雑なSQL*Loader制御ファイルの作成方法を学ぶのに最適です。

結論

SQL*Loaderのエキスプレス・モードは、最も一般的なファイル形式であるCSVファイルのロードに必要な作業を簡略化します。自己最適化プロセスにより、データのロードには可能な限り最速のメカニズムが使用されます。高速ロードをカスタマイズするためのオプション・パラメータが提供されており、複雑な制御ファイルもすばやく作成できます。ファイルを使用してOracleデータベースにデータをロードする必要がある場合は、SQL*Loaderエキスプレス・モードの使用を検討してください。

CONNECT WITH US

+1.800.ORACLE1までご連絡いただくか、oracle.comをご覧ください。

北米以外の地域では、oracle.com/contactで最寄りの営業所をご確認いただけます。



blogs.oracle.com



facebook.com/oracle



twitter.com/oracle

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0120

SQL*Loaderエクスプレス・モードによるOracleデータベースのロード

2021年8月

著者：Jim Stenoish

共著者：Bill Beauregard

