Oracle9i

Oracle実践研修1

ユーティリティ活用

2006.10.4

自己紹介

コミュニケーションテクノロジーズ株式会社
エージェントシステム開発部
畠山基裕
連絡先：hata@comtec.co.jp

[Oracleの経験]

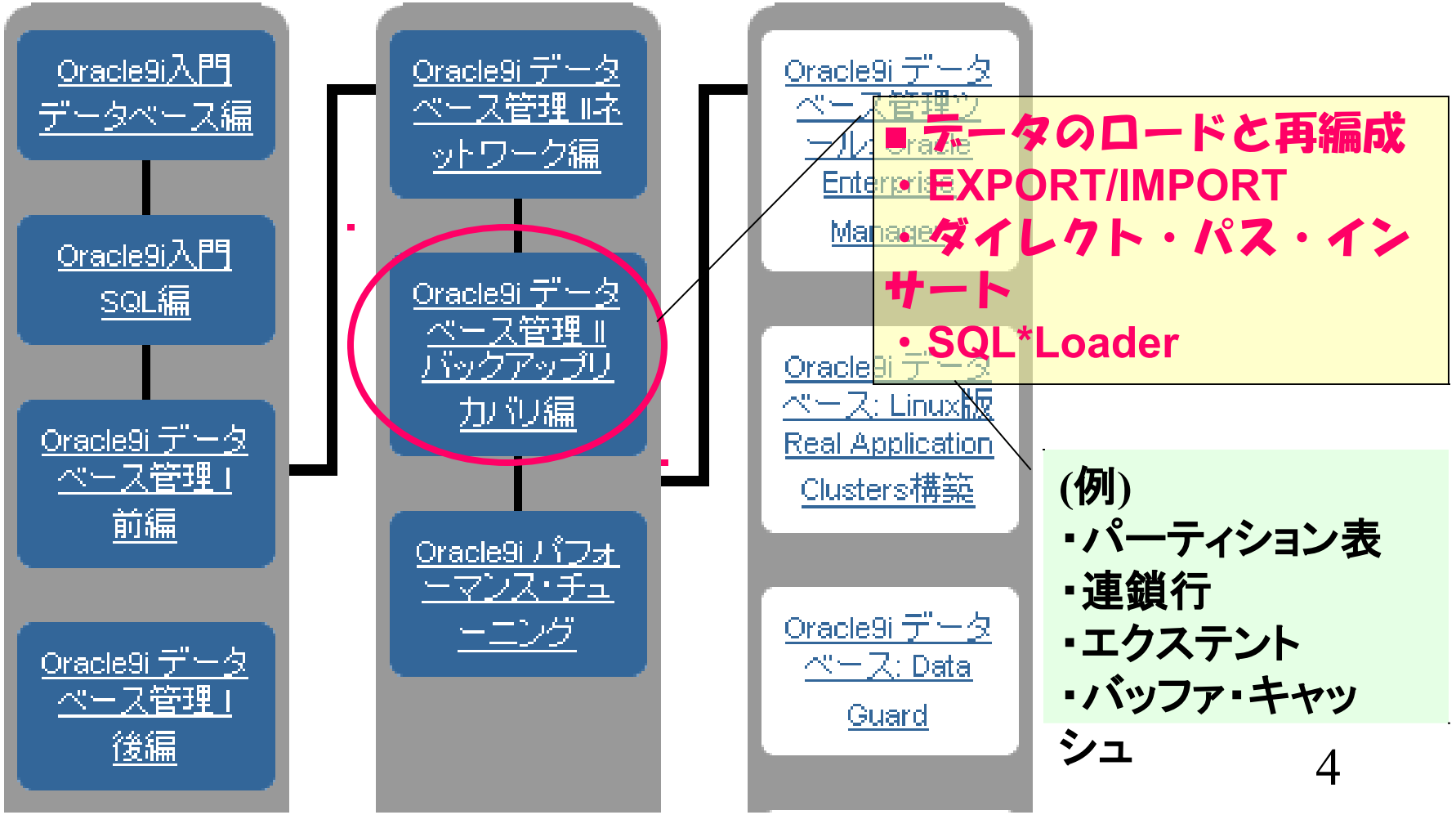
- Oracle Applications(現E-business Suite)を約4年
  - 販売系2つ、生産系3つ
  - セットアップ、機能評価、Extension設計/テスト、チューニング、運用支援
- その他、システム開発

この講座について

- プロ講師ではない
- 独自テキスト
  - 1から10まですべて書いているわけではない
  - 実務経験＋マニュアル＋ネット情報
- マニュアル、ネットから情報を得て自分で解決することが重要
  - OTNの講座やフォーラム、その他

ユーティリティ位置付け

- Oracle9i **データベース管理者ラーニング・パス**



アンケート

- 業務でのOracleデータベース
  - SQL*Plusを使っているか
  - 担当分野：設計／実装／運用管理
  - 最もよく使う、最も新しい・古いバージョン(10g/9i/8i/8/7)
  - 最もよく使うエディション(EE、SE、他)
  - OEMは利用するか
  - パーティション表は利用するか
- 特に現在の業務で困っていること
- OTN
  - 知っているか／登録しているか／活用しているか

実習環境

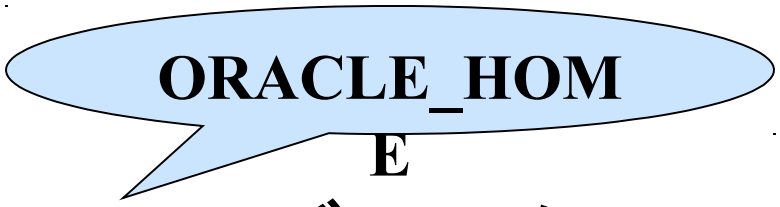
インスタンス名 : smpl

ユーザー/パスワード

: scott/tiger

: oe/oe

: system/system



Oracleディレクトリ : c:\oracle\ora91

実習ディレクトリ : c:\oracle\jisshu

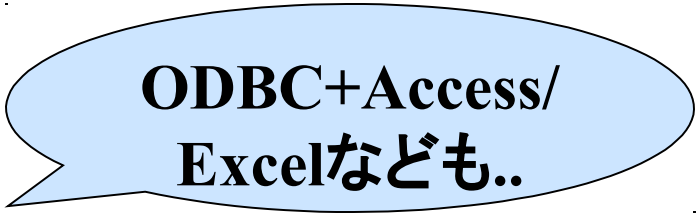
データを表に格納する方法

[本日の内容]

・エクスポート/インポート

・SQL*Loader

・ダイレクト・パス・インサート

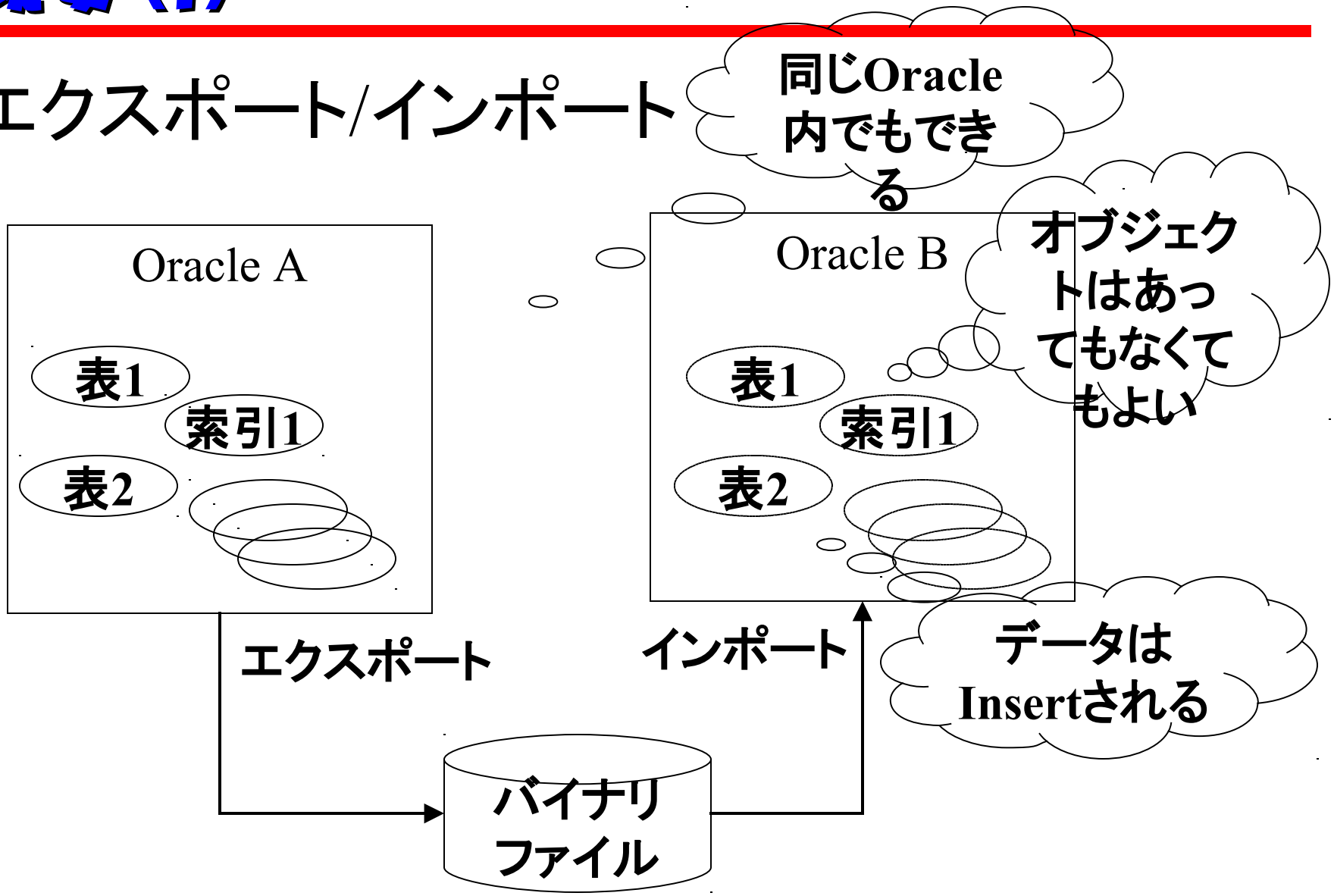


[本日の目的]

・上記ユーティリティを実習し、慣れ、業務で活用できる

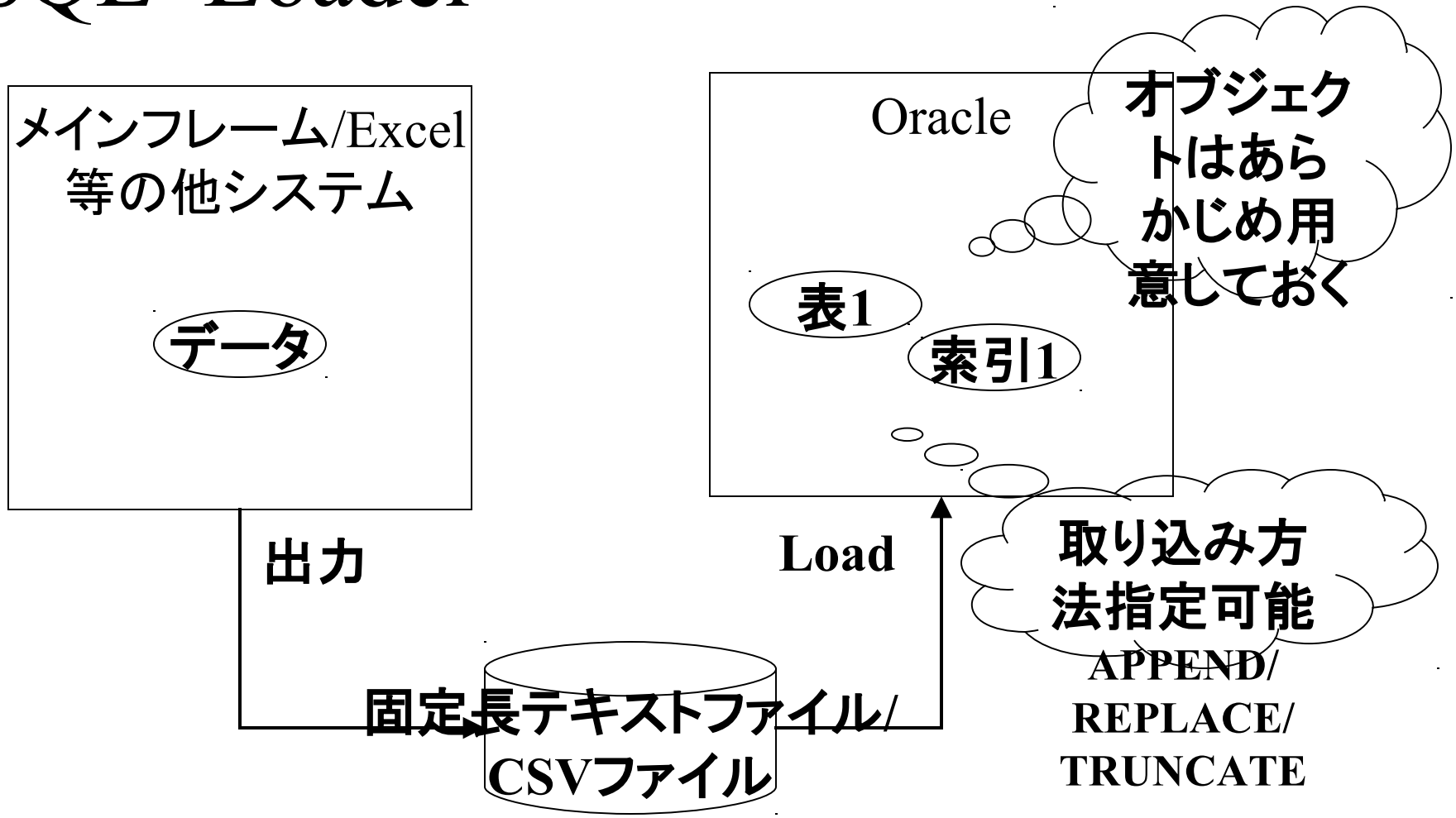
概要(1)

エクスポート/インポート



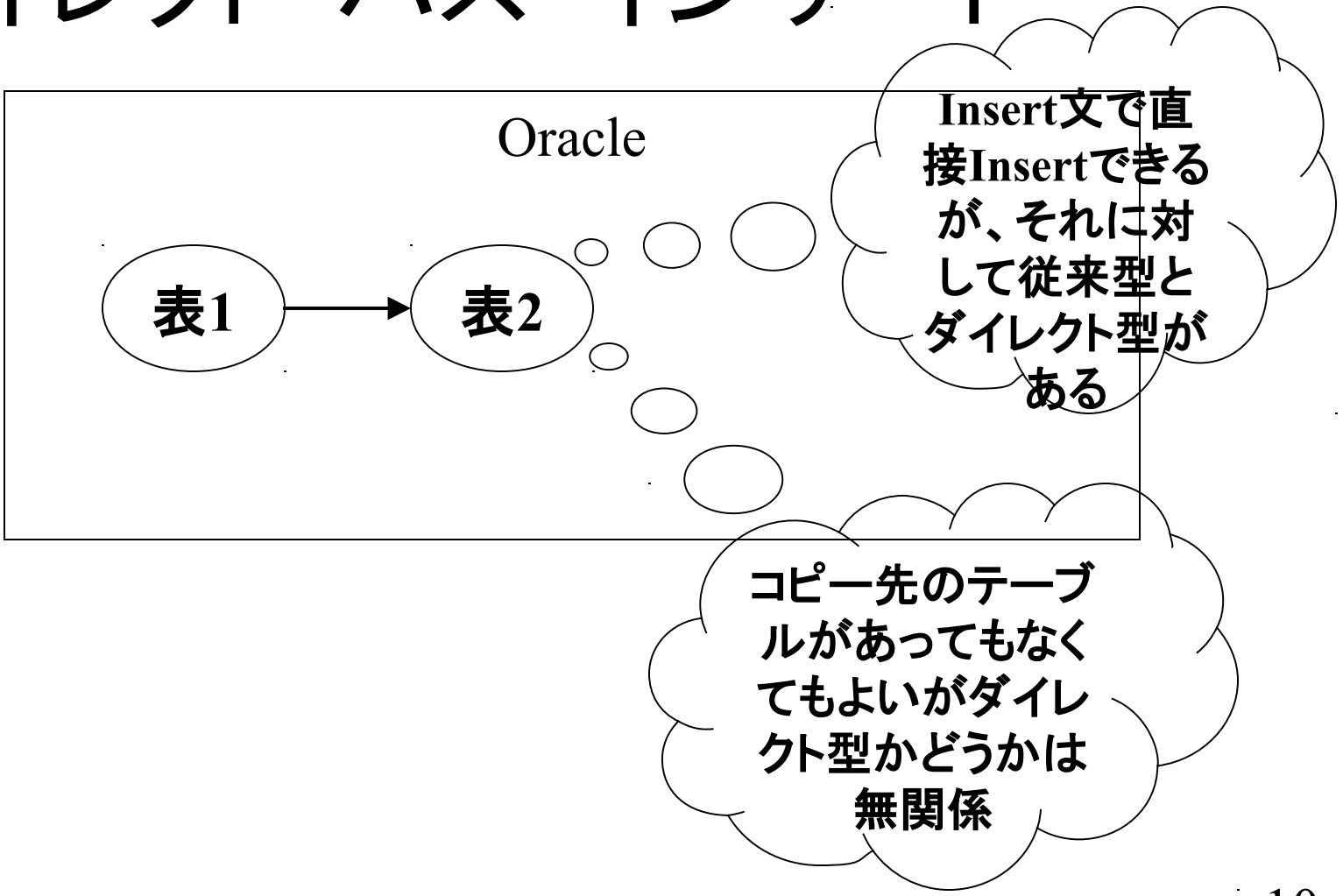
概要(2)

SQL*Loader



概要(3)

ダイレクト・パス・インサート



エクスポート／インポートの特徴

・表定義やデータ(オブジェクト)をOSファイル(バイナリ形式)へ保存し(エクスポート)、それを取り込める(インポート)

・ハードウェア間や異なったバージョンのデータベース間でオブジェクトの移動ができる

・コマンドラインで「exp」「imp」を実行

エクスポート／インポートの活用例

- バックアップ／リカバリー
  - 運用／テスト
- データの移動
  - 本社データベースから支社データベースへ
- データベースの移行
  - Oracle8をOracle9iへ
- オブジェクトの再構成(??)
  - 表、索引の断片化解消

8iからローカル管理表領域がサポートされており、断片化が問題になりにくくなっている。自動セグメント領域管理など。また他にも方法あり。

(参考)移行行と連鎖行

- 移行行
  - 最初は1つのデータ・ブロックに収まっていた行が更新されて行の長さが増え、ブロック空き領域が満杯になった場合、行全体が新しいブロックに収まるよう、行全体のデータが別のデータ・ブロックに移行される。移行行の元の行断片は、移行された行が含まれる新規のブロックをポイントするように保存され、移行された行の行IDは変化しない。
- 連鎖行
  - 行が最初の挿入時に大きすぎて1データ・ブロックに収まらない場合、その行のデータが、複数の鎖状のデータ・ブロックに格納される。データ型LONGまたはLONGRAWの列が含まれるような長い行を処理する場合に、頻繁に発生。
- 問題点
  - その行の情報を取り出すために複数のデータ・ブロックのスキャンが必要になるため、I/Oパフォーマンスが低下する。

(参考) deleteによる断片化

- 行をdeleteしたら？

詳細は、「実践研修2 領域＋インデックス」で。

エクスポートのパラメータ （括弧内はデフォルト）

USERID	ユーザー名／パスワード
BUFFER	データ・バッファ・サイズ
FILE	出力ファイル（EXPDAT.DMP）
LOG	ログファイル
GRANTS	権限のEXPORT（Y）
INDEXES	索引のEXPORT（Y）
ROWS	表データのEXPORT（Y）
CONSTRAINTS	表データに対する制約のEXPORT（Y）
DIRECT	ダイレクトパス（N）
TABLES	指定した表名に関わるオブジェクトをエクスポート
OWNER	ユーザー単位でエクスポート
FULL	全てをエクスポート（N）；EXP_FULL_DATABASE権限
TABLESPACES	表領域単位でエクスポート；9i新機能
PARFILE	パラメータファイル名
QUERY	指定した行だけを抽出；whereで指定

同時使用不可

エクスポートの実習(1)

1. コマンドプロンプトを起動し実習ディレクトリへ移動
   >cd c:\oracle\jisshu
2. 環境変数設定
   >set ORACLE_SID=smpl
3. ヘルプを表示
   >exp help=y
4. 補助資料を参考にいくつかの処理(次頁)を実行
   「Oracle9iデータベース・ユーティリティ」
   J06265-01.pdf → OTNからDL可能（必要部分を配布します）

例)exp userid=scott/tiger tables=emp file=emp.dmp log=emp.log

エクスポートの実習(2)

- scott/tigerでemp表をexport(useridとtablesのみ指定)
- (続き)ログファイル、出力ファイルを指定する
- (続き)パラメータファイルで実行する(useridもこの中に入れられる)
- oe/oeでproduct_descriptionsとinventoriesをexportする
- (続き)パターン一致(“%s”)でテーブルをexportする
- (続き) feedback=100をつけて実行してみる
- ユーザーoeでoeのすべてのオブジェクトをexportする
- (続き) file_format=exp%s filesize=1Mを試す
- (続き) direct=yを試す
- system/systemでscottの全オブジェクトをexportする
- oe/oeでinventoriesのquantity_on_hand>100の行だけをexport

エクスポートその他(1)

- QUERY句 select～where A='a' and B>100なら
  QUERY='where A=''a'' and B^>100'

シングル
コーテーション2

- オブジェクトの再構成
  compress=y 複数Extentが1つにまとまる
- 読み取り一貫性 ; consist=y
- ソート ; query句にorder byを入れる
- パフォーマンスチューニング
  bufferを設定（デフォルトは4096）
  配列(フェッチ)行数×行の最大バイト数

エクスポートその他(2)

•ダイレクトパス; direct=y 高速処理される

従来型パス・エクスポート

SQL のSELECT 文によって、表からデータが抽出される。データはディスクからバッファ・キャッシュに読み込まれ、評価バッファに転送される。式の評価が終了すると、データはエクスポート・クライアントへ転送され、エクスポート・ファイルに書き込まれる。

ダイレクト・パス・エクスポート

データがディスクからバッファ・キャッシュに読み込まれ、行がエクスポート・クライアントに直接転送されるため、従来型に比べて非常に高速。評価バッファはバイパスする。データは、すでにエクスポート・ユーティリティが要求する形式になっているため、不要なデータ変換をする必要がない。データはエクスポート・クライアントに転送され、ファイルに書き込まれる。

制限

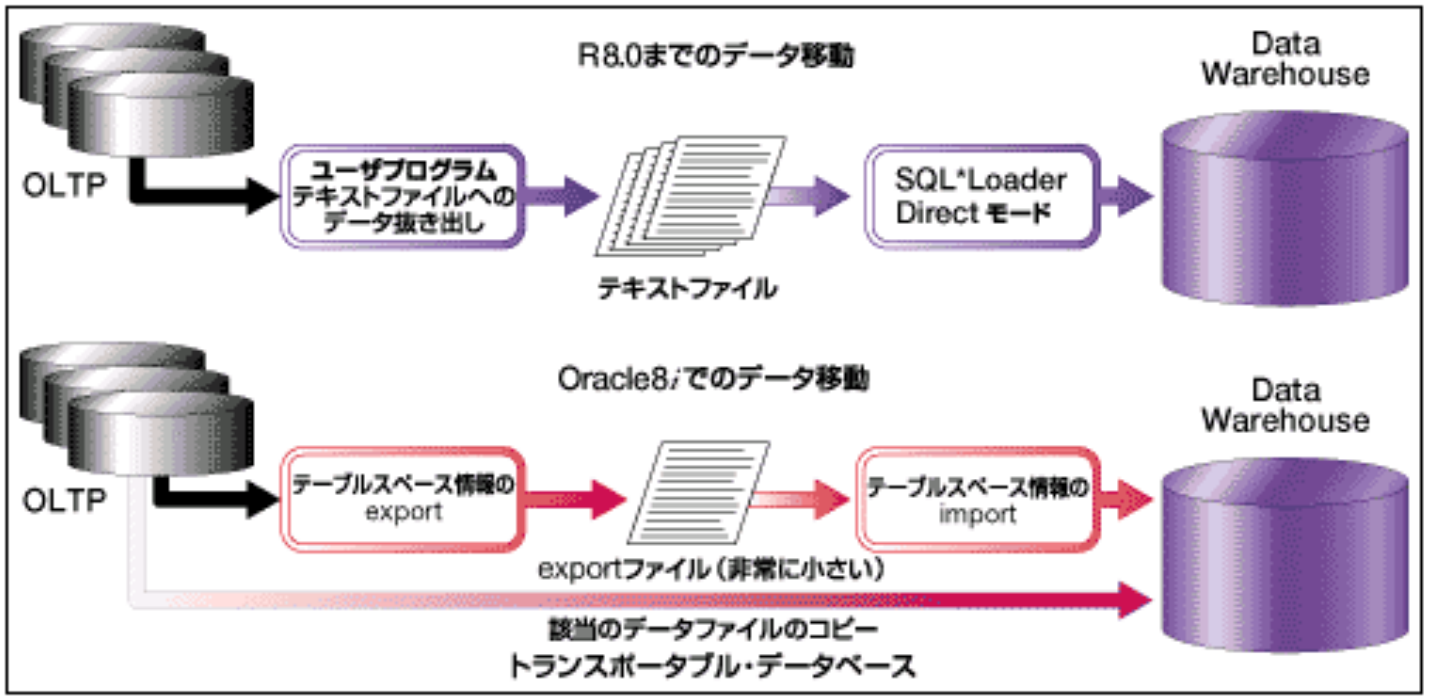
BUFFER(RECORDLENGTHで代替)、QUERYは使用不可、LOBカラムを持つ表では自動的に"Y"に設定

エクスポートその他(3)

- Disk Sizeの確認
  - 出力に十分な容量
  - SELECT SUM(BYTES) FROM USER_SEGMENTS WHERE SEGMENT_TYPE='TABLE';
- ダイレクトパスのパフォーマンス(RECORDELENGTH)
  - DB_BLOCK_SIZE、I/Oブロックサイズの倍数
  - DBと出力先のディスク・ドライブ

エクスポートその他(4)

- トランスポータブル表領域
  - 同一OSの別システム間で表領域(データファイル)をコピー
  - EXP TRANSPORT_TABLESPACE=y TABLESPACES=(sales_1,sales_2)
  - メタデータのexp/imp＋データファイルのコピー



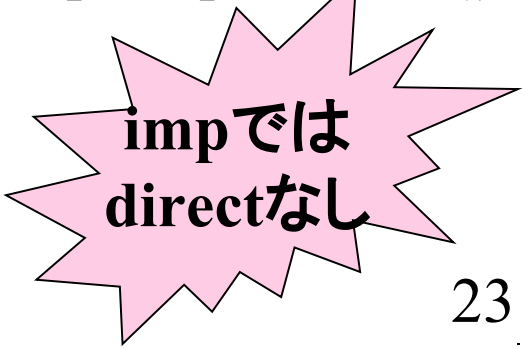
インポートの順序

1. 型定義
2. 表定義(create table)
3. 表データ(insert)
4. 表索引(create index)
5. 整合性制約、ビュー、プロシージャおよびトリガー
6. ビットマップ索引、ファンクション索引およびドメイン索引

インポートのパラメータ

（括弧内はデフォルト）

USERID	ユーザー名／パスワード
BUFFER	データ・バッファ・サイズ
FILE	入力ファイル（EXPDAT.DMP）
LOG	ログファイル
SHOW	EXPORTファイルの内容表示（N）
IGNORE	作成時エラー無視（N）; 既存表にimportする場合
GRANTS	権限のIMPORT（Y）
INDEXES	索引のIMPORT（Y）
ROWS	表データのIMPORT（Y）
FULL	全インポートモード（N）; IMP_FULL_DATABASE権限
FROMUSER	所有するユーザー
TOUSER	IMPORT先のユーザー
TABLES	表名
TABLESPACES	表領域



インポートに必要な権限

CREATE TABLE

CREATE SEQUENCE

CREATE TRIGGERなど

上記の権限はRESOURCEロールに全て含まれている。

```
SQL> select * from user_role_privs;

USERNAME                       GRANTED_ROLE                   ADM DEF OS_
------------------------------ ------------------------------ --- --- ---
SCOTT                          CONNECT                        NO  YES NO
SCOTT                          RESOURCE                       NO  YES NO
```

インポートの実習(1)

1. コマンドプロンプトを起動し実習ディレクトリへ移動
   >cd c:\oracle\jisshu
2. 環境変数設定
   >set ORACLE_SID=smpl
3. ヘルプを表示
   >imp help=y
4. 次のいくつかの処理を実行する

例)imp userid=scott/tiger tables=emp

インポートの実習(2)

- scott/tigerでempをexport
- scott/tigerでempをimport、ignore=nで実行する(エラー発生)
- ignore=yで実行する(一意制約でエラー)
- sql*plusを起動しscott/tigerでログイン後、empのレコードを削除し、ignore=yでimport。レコード確認。
- テーブルをdropし、ignore=nでimport。レコード確認。
- scott/tigerのempをoeにimport(userid=oe/oe constraints=y fromuser=scott touser=oe)、外部制約作成エラー(deptがないため)が出るがレコードはimportされる→sql*plusで確認(oe/oe)、empをdropする
- constraints=nで実行、sql*plusで確認→empをdrop

データベース全体の断片化の解消

1. データベース全体のバックアップを取るために、全データベース・エクスポート（FULL=y）を実行。
2. すべてのユーザーがログオフしてから、Oracle データベース・サーバーを停止。
3. データベースを削除。
4. CREATE DATABASE 文を使用して、データベースを再作成。
5. データベース全体をリストアするために、全データベース・インポート（FULL=y）を実行。

※sysユーザのトリガーはexpされないのでimp後手動作成が必要→いろいろな制約に注意!!

表の断片化の解消

1. 表をエクスポート
2. 表を削除
3. 表をインポート

- ただし、表の生成パラメータに問題がある場合（連鎖行が多いなど）は、問題を解消できるパラメータで表を再作成し、レコードのみをロードする。

インポートその他

- エクスポートした内容すべてをインポートする場合は、FULL=Yを指定(TABLES=(A,B,C,D)としてエクスポートした場合、同じ指定でインポートしてもよいが、FULL=Yですべてが対象)
- エクスポートした内容の中から、特定の表のみインポート可
- インポートするときは参照整合性違反に注意する
  - –既存の表では手動で使用禁止にするとよい
    ALTER TABLE "SCOTT"."EMP" DISABLE CONSTRAINT "FK_DEPTNO"
  - –通常、参照制約はすべての表の後にインポート
  - –実習の例(scottからoeへ)では、empとともにdeptもエクスポート/インポートすればよい

Export/Importによる運用バックアップ

全体エクスポート、累積エクスポート、増分エクスポートを利用して、1ヶ月のバックアップスケジュールを作成する。

日	月	火	水	木	金	土
			1 全体	2 増分	3 増分	4 累積
5 増分	6 増分	7 増分	8 増分	9 増分	10 増分	11 累積
12 増分	13 増分	14 増分	15 増分	16 増分	17 増分	18 累積
19 増分	20 増分	21 増分	22 増分	23 増分	24 増分	25 累積
26 増分	27 増分	28 増分	29 増分	30 増分	31 増分	

月初に全体エクスポートをする。

週末に増分エクスポートをする。

その他の日は増分のエクスポートをする。

8iまでの機能
(INCTYPE)

バックアップは他のツール（RMAN等）で行う！

Export/Importによる運用バックアップ

21日に障害が発生した場合の回復手順

日	月	火	水	木	金	土
			1 全体	2 増分	3 増分	**4 累積**
5 増分	6 増分	7 増分	8 増分	9 増分	10 増分	**11 累積**
12 増分	13 増分	14 増分	15 増分	16 増分	17 増分	**18 累積**
19 増分	**20 増分**	21				

8iまでの機能
(INCTYPE)

1. エクスポートファイルの情報をロードするために最新のファイルをインポートする。

   Imp system/manager INCTYPE=SYSTEM FULL=Y FILE=DAY_19.DMP

2. 全体エクスポートファイルをインポートする。

   Imp system/manager INCTYPE=RESTORE FULL=Y FILE=DAY_1.DMP

3. 累積エクスポートファイルを全てインポートする。

   Imp system/manager INCTYPE=RESTORE FULL=Y FILE=DAY_4.DMP

   Imp system/manager INCTYPE=RESTORE FULL=Y FILE=DAY_11.DMP

   Imp system/manager INCTYPE=RESTORE FULL=Y FILE=DAY_18.DMP

4. 増分エクスポートファイルを全てインポートする。

   Imp system/manager INCTYPE=RESTORE FULL=Y FILE=DAY_19.DMP

   Imp system/manager INCTYPE=RESTORE FULL=Y FILE=DAY_20.DMP

ＳＱＬ＊Ｌｏａｄｅｒの特徴

外部で作成したデータをORACLEのデータベースに取り込むことができる。

入出力ファイルの種類

- 入力ファイル

データファイル：ロードされるデータ(固定長/CSVなど)

制御ファイル　：ロードされる表、入力データ、フィールド仕様など

パラメーター　：コマンド行パラメータ

- 出力ファイル

ログファイル　：データロードした結果の詳細

廃棄ファイル　：ロードの対象とならないデータ(WHEN句の対象外のデータ)

不良ファイル　：エラーが発生してロードされなかったデータ

データファイルの例(1)

固定長データ

```
7369SMITH      CLERK     79021980/12/17   800.00          20
7499ALLEN      SALESMAN  76981981/02/20  1600.00   300.0030
7521WARD       SALESMAN  76981981/02/22  1250.00   500.0030
7566JONES      MANAGER   78391981/04/02  2975.00          20
7654MARTIN     SALESMAN  76981981/09/28  1250.00  1400.0030
7698BLAKE      MANAGER   78391981/05/01  2850.00          30
7782CLARK      MANAGER   78391981/06/09  2450.00          10
7788SCOTT      ANALYST   75661982/12/09  3000.00          20
7839KING       PRESIDENT      1981/11/17  5000.00          10
7844TURNER     SALESMAN  76981981/09/08  1500.00     0.0030
7876ADAMS      CLERK     77881983/01/12  1100.00          20
7900JAMES      CLERK     76981981/12/03   950.00          30
7902FORD       ANALYST   75661981/12/03  3000.00          20
7934MILLER     CLERK     77821982/01/23  1300.00          10
```

制御ファイルの構文

・固定長データの場合

LOAD DATA

INFILE 'データファイル名'

モード指定

INTO TABLE テーブル名

(

フィールド名　POSITION(データ位置)　データ型,

フィールド名　POSITION(データ位置)　データ型,

フィールド名　POSITION(データ位置)　データ型

)

制御ファイルの構文(2)

モード指定について

- APPEND
  - –表にデータを追加する
- REPLACE
  - –表のデータを削除(delete)して、新規にデータを挿入する;deleteトリガーは実行される
- TRUNCATE
  - –表のデータを削除(truncate)して、新規にデータを挿入する;参照整合性が使用禁止になっている必要あり

制御ファイルの例

・固定長データ用制御ファイルの例

```
LOAD DATA
INFILE 'emp.dat'
APPEND
INTO TABLE emp
(
   empno          POSITION(01:04)        INTEGER EXTERNAL,
   ename          POSITION(05:14)        CHAR,
   job            POSITION(15:23)        CHAR,
   mgr            POSITION(24:27)        INTEGER EXTERNAL,
   hiredate       POSITION(28:37)        DATE "YYYY/MM/DD",
   sal            POSITION(38:45)        DECIMAL EXTERNAL,
   comm           POSITION(46:53)        DECIMAL EXTERNAL,
   deptno         POSITION(54:55)        CHAR
)
```

文字型の数値=通常のテキストファイルでは必ず指定

データファイルの例(2)

可変長データ

7369,SMITH,CLERK,7902,1980/12/17,800,,20

7499,ALLEN,SALESMAN,7698,1981/02/20,1600,300,30

7521,WARD,SALESMAN,7698,1981/02/22,1250,500,30

7566,JONES,MANAGER,7839,1981/04/02,2975,,20

7654,MARTIN,SALESMAN,7698,1981/09/28,1250,1400,30

7698,BLAKE,MANAGER,7839,1981/05/01,2850,,30

7782,CLARK,MANAGER,7839,1981/06/09,2450,,10

7788,SCOTT,ANALYST,7566,1982/12/09,3000,,20

7839,KING,PRESIDENT,,1981/11/17,5000,,10

7844,TURNER,SALESMAN,7698,1981/09/08,1500,0,30

7876,ADAMS,CLERK,7788,1983/01/12,1100,,20

7900,JAMES,CLERK,7698,1981/12/03,950,,30

7902,FORD,ANALYST,7566,1981/12/03,3000,,20

7934,MILLER,CLERK,7782,1982/01/23,1300,,10

制御ファイルの構文

・可変長データの場合

LOAD DATA

INFILE 'データファイル名'

モード指定

INTO TABLE テーブル名

FIELDS TERMINATED BY '区切り文字' OPTIONALLY ENCLOSED BY '囲み文字'

TRAILING NULLCOLS

(

フィールド名,

フィールド名,

フィールド名

)

制御ファイルの例

・可変長データ用制御ファイルの例

```
LOAD DATA
INFILE 'emp.dat'
APPEND
INTO TABLE emp
FIELDS TERMINATED BY ','
TRAILING NULLCOLS
(
    empno,
    ename,
    job,
    mgr,
    hiredate DATE "YYYY/MM/DD",
    sal,
    comm,
    deptno
)
```

ＳＱＬ＊Ｌｏａｄｅｒのパラメータ

USERID	データベースに接続するユーザー名／パスワード
CONTROL	制御ファイル名
DATA	データファイル名(*制御ファイル内に指定可)
PARFILE	パラメータファイル名
LOG	ログファイル名
BAD	不良ファイル名(*)
DISCARD	廃棄ファイル名(*)
ERRORS	最大不良レコード数
DIRECT	ダイレクトパスロード指定
PARALLEL	パラレルダイレクトパスロード指定

従来型パス・ロードとダイレクト・パス・ロードの特徴

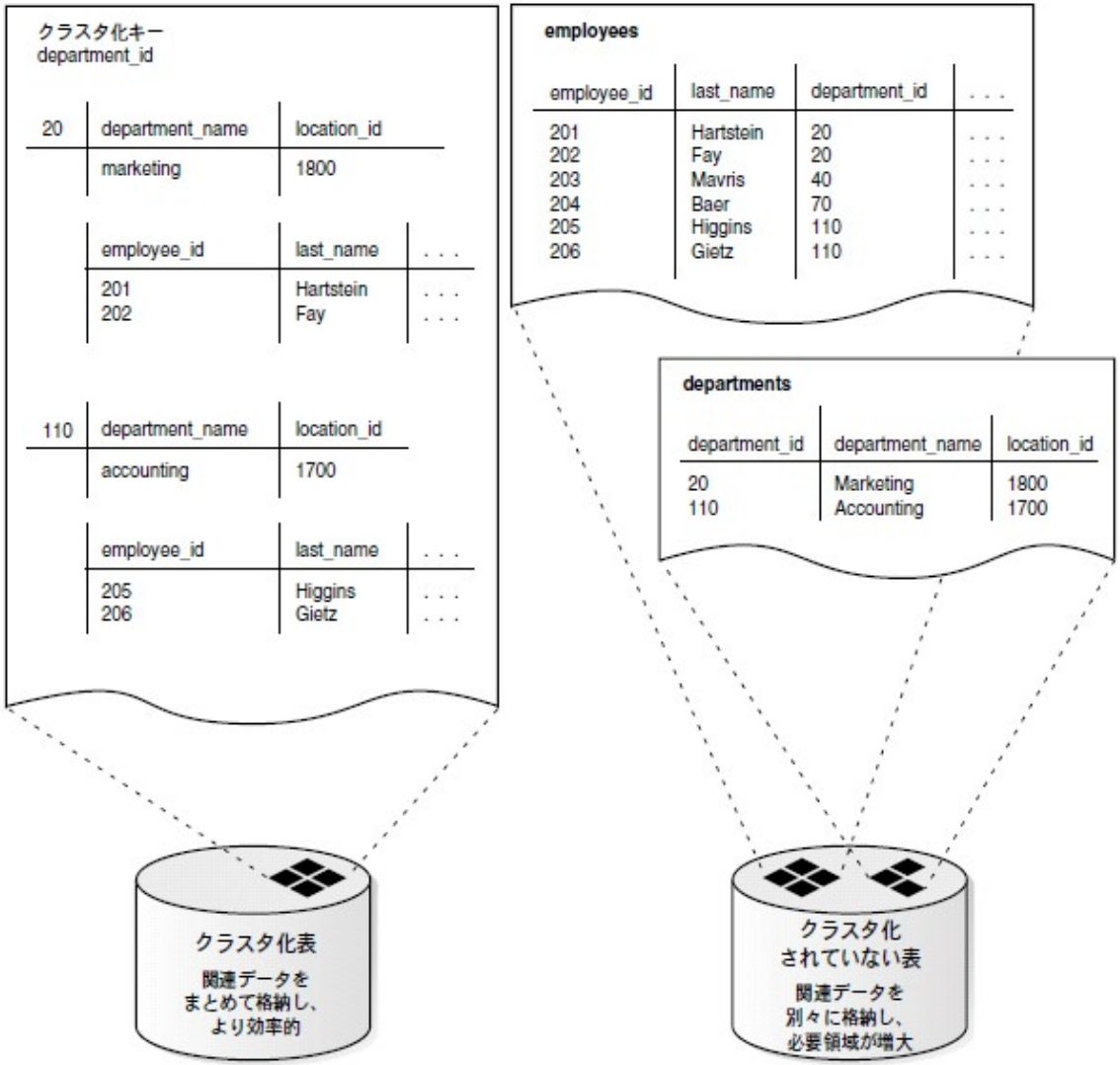
- 従来型
  - –表に対してinsert文が実行される、トリガーも起動される
- ダイレクト
  - –Oracle データ・ブロックをフォーマットし、データ・ブロックを直接データ・ファイルに書き込むので高速
  - –Insertトリガーは処理前に使用不可となるが処理後使用可に戻る。ただし適用はされない。
  - –制限
    - –表がクラスタ化されていないこと
    - –ロードする表に未処理のアクティブ・トランザクションがないこと
    - –VARRAY のロード、親表を子表とともにロード、BFILE 列のロードはできない
    - –参照整合性制約は処理前に使用不可となるので後で戻すこと

emp表とdept表のdeptno列を同じブロックに格納

従来型パス・ロードとダイレクト・パス・ロードの特徴

- 補足

クラスタ化された表



従来型パス・ロードとダイレクト・パス・ロードの特徴

- 補足

VARRAY

–オブジェクト・データ型の配列

–CREATE TYPE prices AS VARRAY(10) OF NUMBER(12,2);
prices型のVARRAYで、最大10 個までの要素を入れることができ、各要素のデータ型はNUMBER(12,2)

BFILE

–LOB（ラージオブジェクト）データ型

–構造化されていないバイナリ・データを、データベースの外にあるオペレーティング・システムのファイルに格納。BFILE 列または属性は、データが含まれる外部ファイルを参照するファイル・ロケータを格納する。

SQL*Loaderの実行（従来型）

```
sqlldr userid=scott/tiger control=emp.ctl
       log=emp.log bad=emp.bad
```

SQL*Loaderの実行（ダイレクト）

```
sqlldr userid=scott/tiger control=emp.ctl
        log=emp.log bad=emp.bad direct=true
```

どちらを利用するか？

- **ダイレクト・パス**

  短時間で大量のデータをロードする必要があるとき。

  既に存在しているデータに比べてかなり大量なデータをロードするとき。

- **従来型パス**

  ロードと同時に、索引付の表へのアクセス(selectも)や、索引なし表への挿入・更新が必要なとき。

  巨大な索引付きの表に対する少量のロードを行うとき。

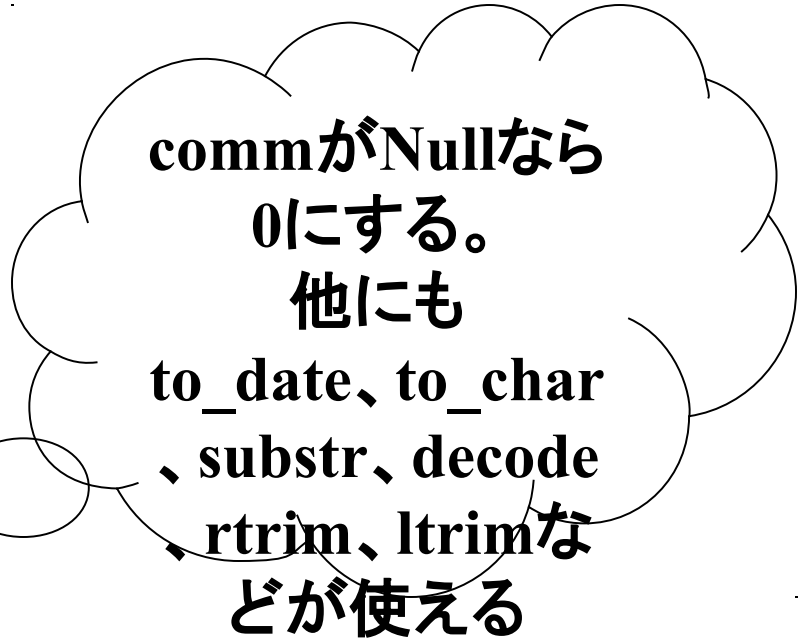
  トリガーや参照整合性を実行させたいとき。

  削除による空き領域が多い表へのロードを行うとき。

特殊な制御ファイルの例(1)

・SQL関数を利用した例

```
LOAD DATA
INFILE 'emp.dat'
APPEND
INTO TABLE emp
FIELDS TERMINATED BY ','
TRAILING NULLCOLS
(
    empno,
    ename,
    job,
    mgr,
    hiredate DATE "YYYY/MM/DD",
    sal,
    comm "NVL(:comm,0)",
    deptno
)
```



特殊な制御ファイルの例(2)

・制御ファイルにロードするデータを記入する例

```
LOAD DATA
INFILE *
APPEND
INTO TABLE dept
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
TRAILING NULLCOLS
( deptno, dname, loc )
BEGINDATA
10,"ACCOUNTING","NEW YORK"
20,"RESEARCH","DALLAS"
30,"SALES","CHICAGO"
40,"OPERATIONS","BOSTON"
```

特殊な制御ファイルの例(3)

1 50 Manufacturing — DEPT record

2 1119 Smith 50 — EMP record

2 1120 Snyder 50

1 60 Shipping

レコードID フィールドで区別される。

```
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)
```

特殊な制御ファイルの例(4)

1行に2レコード

1119 Smith 1120 Yvonne

1121 Albert 1130 Thomas

```
INTO TABLE emp
(empno POSITION(1:4) INTEGER EXTERNAL,
ename POSITION(6:15) CHAR)
INTO TABLE emp
(empno POSITION(17:20) INTEGER EXTERNAL,
ename POSITION(21:30) CHAR)
```

SQL*Loaderのためのデータ抽出方法

emp表データ抽出用SQLの例

```
set pages 0
set lines 60
select empno  ||','||
       ename ||','||
       job  ||','||
       mgr ||','||
       to_char(hiredate, 'yyyy/mm/dd')||','||
       sal||','||
       comm||','||
       deptno
from emp
;

exit
```

SQL*Loader実習

- sql*plusにscott/tigerでログインしemp表からテキストデータを作る
- 制御ファイルを作る
- emp表のレコードをdeleteする
- SQL*Loaderでロードし、内容を確認する
- emp表のレコードからempno=7521以外を削除する
- SQL*Loderで再びロードする
- emp表のレコードからempno=7521以外を削除する
- into table emp when empno != '7521'でロード

SQL*Loader実習

- （余裕があれば）バイナリデータのロード
  - テーブル作成(navis1_tbl_create.sql)
  - ロード(jpgload.ctl)

ダイレクト・パス・インサート

表のコピーをする際に、バッファキャッシュ(SGAの一部)を経由せずに直接データファイルへ書き込みを行い、処理を高速化する手法。

ただし、表には排他ロックがかかるため、その表を他の処理で使用できない。

従来の方法は、空領域を再利用するが、ダイレクトの場合は再利用せず既存データの後ろに挿入する。

SQL*Loaderのダイレクトも同じ

ダイレクトパスインサートの構文

シリアルモードの場合~EnterpriseEditionでない場合は必ずシリアル

構文

insert /*+APPEND */ into 表名 select文;

例）SQL> insert /*+append */ into new_dept select * from dept;

ダイレクトパスインサートの補足

EnterpriseEdition

-シリアルモード/パラレルモード

EnterpriseEditionでパラレルモードにする場合

パラレルモードは1つのSQLを複数のプロセスに分割して実行する。

ALTER SESSION ENABLE PARALLEL DML;

パラレルモードではダイレクトパスがデフォルトになる(通常パスにするにはNOAPPENDを指定)。ターゲット表にパラレル属性が指定されていることがダイレクトの条件(APPENDをつければ可)。

シリアルにするにはDISABLE PARALLEL DMLとする。

ダイレクトパスインサートの実習

- SQL*Plusにoe/oeでログインする
- create table orders_test1 as select * from orders;
- orders_test1のレコードを削除
- appendヒントを使ってダイレクトパスインサートによりordersの内容をorders_test1にコピーする。→commit前後にselectしてみる。
- orders_test1のレコードを削除
- create table orders_test2 as select * from orders;
- orders_test2のレコードを削除
- セッションをパラレルモードに変更する
- appendヒントを使わずにordersの内容をorders_test2にコピー。
- orders_test2のレコードを削除

GUIでのエクスポート/インポート

- Enterprise Managerから実行できる
- EM管理者でなければ駄目
- セットアップが必要
  - EMのRepository
  - サービス(ManagementServer、Agent)
  - ノードの構成
- 紹介のみ

時間があれば
ODBCも紹介

参考情報

- **OTN(Oracle Technology Network)**

  http://otn.oracle.co.jp/
  登録すればマニュアル(pdf)がダウンロードできる

- **Oracle東北支社の無料セミナー**

  http://www.oracle.co.jp/corp/tohoku/events/index.html