

Oracle Direct Seminar



ORACLE®

画像・ドキュメントもDBで管理する
～ LOB取り扱い説明書 ～

日本オラクル株式会社

Oracle Direct



Agenda

- LOB概要
- Oracle Database のLOB
- LOBの管理
- LOBの操作
- LOBを扱う上での留意点
- 事例



無償技術サービスOracle Direct Concierge

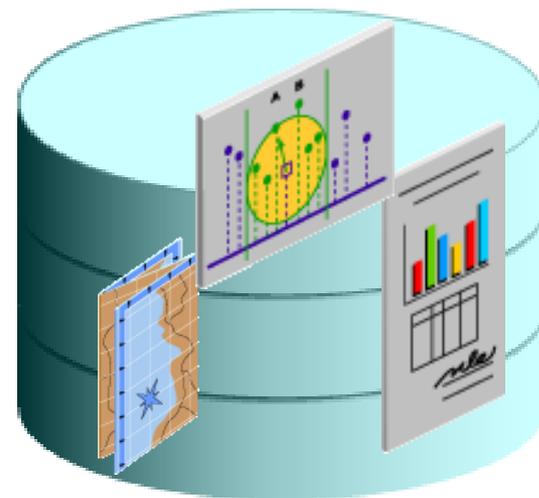
- SQL Serverからの移行アセスメント
 - MySQLからの移行相談
 - PostgreSQLからの移行相談
 - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
 - Oracle Developer/2000 Webアップグレード相談
 - パフォーマンス・クリニック
- Oracle Database 構成相談
- Oracle Database 高可用性診断
 - システム連携アセスメント
 - システムセキュリティ診断
 - 簡易業務診断
 - メインフレーム資産活用

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

ラージ・オブジェクト(LOB)とは

- Oracle Databaseではバージョン8より利用可能
 - SQL99で標準のデータ型として定義されています
- 大量のデータを保持するように設計されたデータ型
 - 最大サイズは128TB
- LONG、LONG RAWに比べると柔軟
 - 事実上、サイズ上限を気にしなくとも良い
 - 1つの表は複数のLOBを利用可能
 - LOBはランダム・アクセス可能
 - LOBオブジェクトの属性になれる
 - 例: DICOMデータ型



ラージ・オブジェクトを使用する理由（その1）

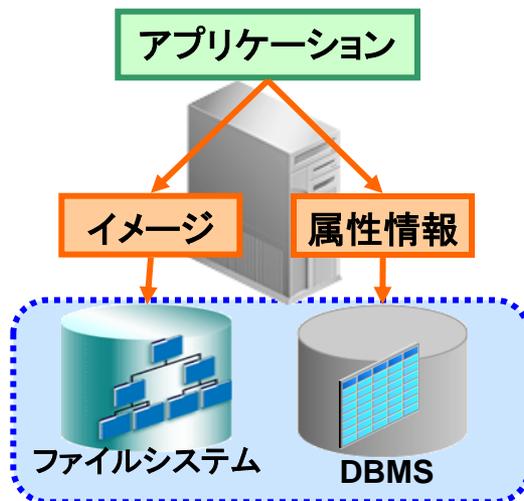
- 現在のアプリケーションでは、次の種類のデータを扱います

データの種類	データ内容	RDB向き？
単純な構造化データ	「表」形式で表現可能	◎
複雑な構造化データ	「オブジェクト」で表現	○
半構造化データ	DBMSで解析されない構造を持つ (XML、PDF、ビジネス文書)	△
非構造化データ	これ以上小さな構造に分解されない (画像データ等)	△

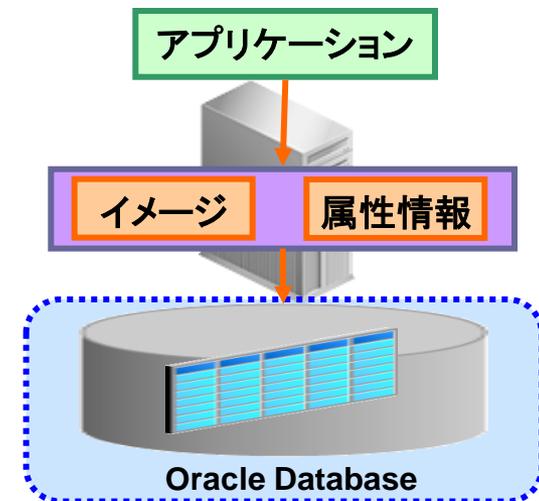
- ラージ・オブジェクトを使用することで、全てのデータを効率良く管理できます

ラージ・オブジェクトを使用する理由（その2）

- すべてのデータを一元管理することのメリット
 - アプリケーション・ロジックが単純化できる
 - 一つのトランザクションで操作が可能に
 - バックアップなどの運用を一元化できる



バックアップ対象: ファイルとDBMS



バックアップ対象: Oracle Database のみ

Agenda

- LOB概要
- Oracle Database のLOB
- LOBの管理
- LOBの操作
- LOBを扱う上での留意点
- 事例



無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
 - MySQLからの移行相談
 - PostgreSQLからの移行相談
 - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
 - Oracle Developer/2000 Webアップグレード相談
 - パフォーマンス・クリニック
- Oracle Database 構成相談
- Oracle Database 高可用性診断
 - システム連携アセスメント
 - システムセキュリティ診断
 - 簡易業務診断
 - メインフレーム資産活用

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

Oracle Databaseで扱うLOB（その1）

- 内部LOB

- Oracle Database内部に格納するLOB

BLOB : バイナリデータを格納

CLOB : 文字データを格納

NCLOB : 各国語キャラクタ・セットの文字データを格納

- 外部LOB

- Oracle Database 表領域の外にあるOSファイルに格納される

BFILE : 読み取り専用(アプリケーション内で書込みできない)

DBMSのトランザクションには関係ない

OSの制限(ファイルサイズ等)

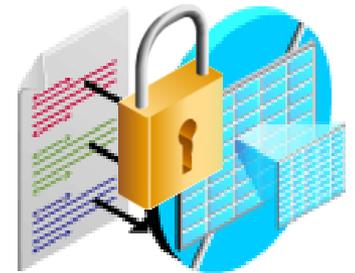
本セミナーでは、内部LOB中心にお話します

Oracle Databaseで扱うLOB（その2）

- 内部LOBには以下の2種類があります
- 永続LOB
 - Oracle Databaseの表の行に存在するLOB
 - トランザクションと関連（コミット、ロールバック可能）
 - 障害が発生してもリカバリ可能
- 一時LOB
 - アプリケーション内部で一時的に利用するLOB
 - セッション終了、明示的解放、スコープ外となるまでアプリケーションに存在

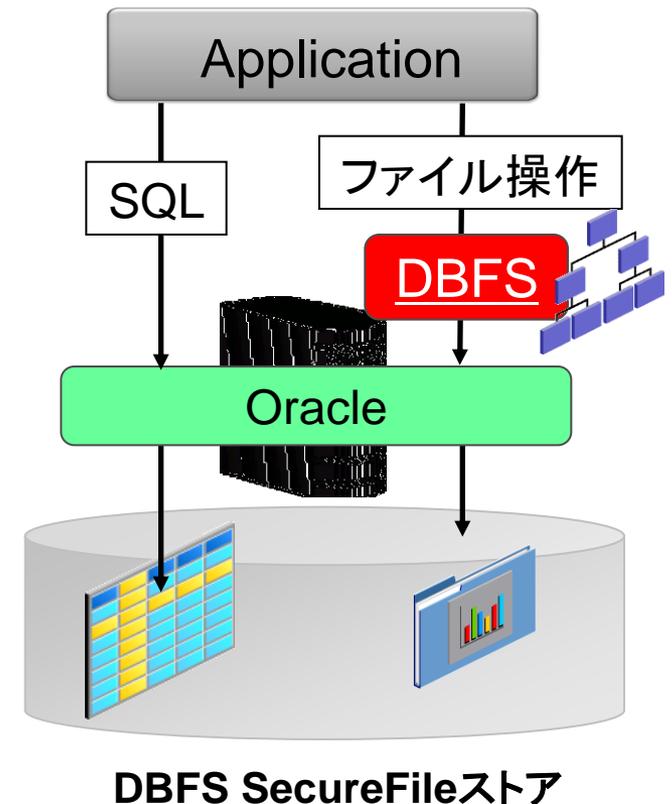
SecureFiles (11g R1~)

- Oracle Database 11g Release 1 の新機能
- SecureFiles は、次の要件を満たすために導入された新世代の LOB データ型です
 - 機能拡張
 - 高パフォーマンス
 - 使いやすさ
- 新しい LOB と従来の LOB は選択可能
 - 新しい LOB データ型を「SECUREFILE」、
従来の LOB データ型を「BASICFILE」と呼びます



Database File System (11g R2~)

- DBFS SecureFileストア
 - Oracle DatabaseのSecureFile LOBに格納されたファイルに対し、ファイルシステムよりアクセス可能とする機能
 - Linux x86 / Linux x86-64 のみ
- DBFS 階層ストア
 - RMANでサポートされているテープや外部ストレージ・デバイスのコンテンツおよびデータベース内の関連するメタデータ(またはプロパティ)を格納
 - Amazon S3



Agenda

- LOB概要
- Oracle Database のLOB
- ➔ • LOBの管理
- LOBの操作
- LOBを扱う上での留意点
- 事例



無償技術サービスOracle Direct Concierge

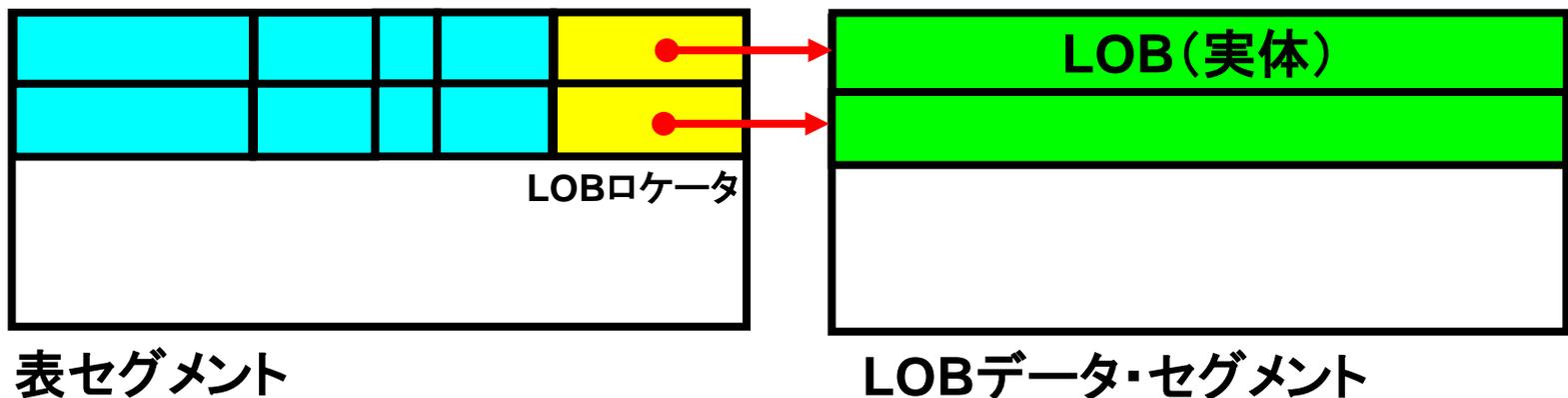
- SQL Serverからの移行アセスメント
 - MySQLからの移行相談
 - PostgreSQLからの移行相談
 - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
 - Oracle Developer/2000 Webアップグレード相談
 - パフォーマンス・クリニック
- Oracle Database 構成相談
- Oracle Database 高可用性診断
 - システム連携アセスメント
 - システムセキュリティ診断
 - 簡易業務診断
 - メインフレーム資産活用

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

LOBロケータ

- LOBが物理的に格納されている場所への参照
 - OS上のファイルをアクセスするアプリケーションで利用するファイルポインタに相当
- 表セグメントに格納されるのはLOBロケータ
 - LOBを操作するにはLOBロケータを用いて専用のAPIを利用
 - SELECTで取得できるのはLOBロケータ



LOBの格納方式

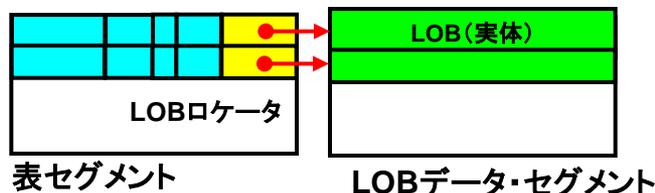
- LOBのインライン格納 (ENABLE STORAGE IN ROW)

- 表と同じセグメントにLOBを格納する方式
- LOBのサイズが4000Byteを越えるとLOBセグメントに格納
- 小さいLOBは追加I/Oが不要
- デフォルトはインライン格納



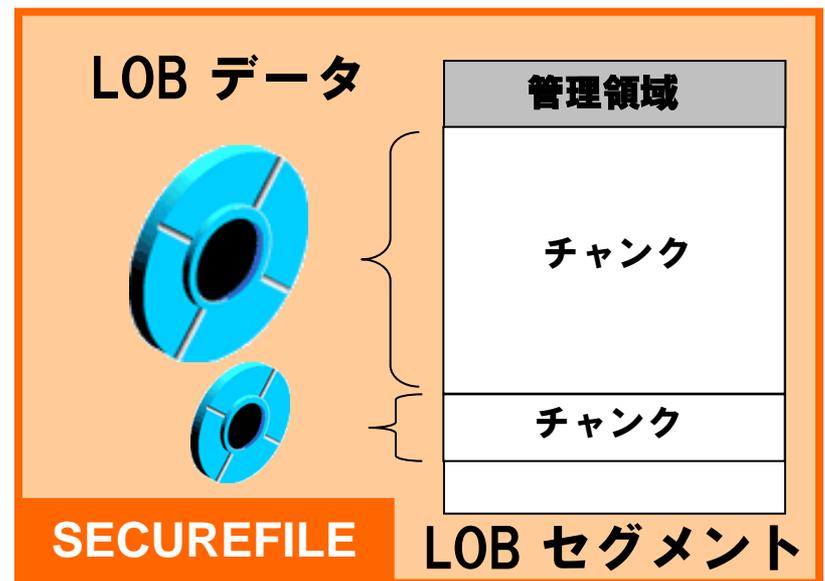
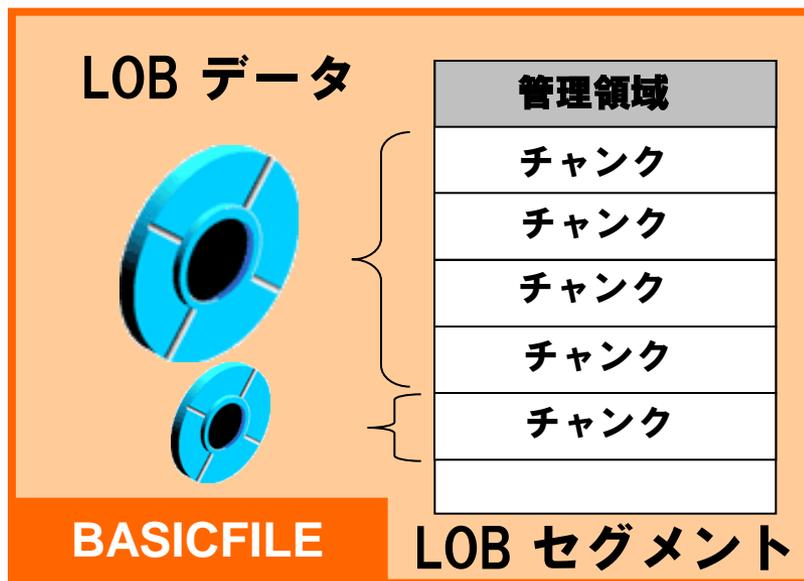
- LOBのアウトライン格納 (DISABLE STORAGE IN ROW)

- LOBのサイズに関わらず、LOBデータをLOBセグメントに格納
- 領域管理が行い易い
- 全表スキャンやレンジ・スキャンが多く発生する表ではアウトライン格納の利用が望ましい



チャンク

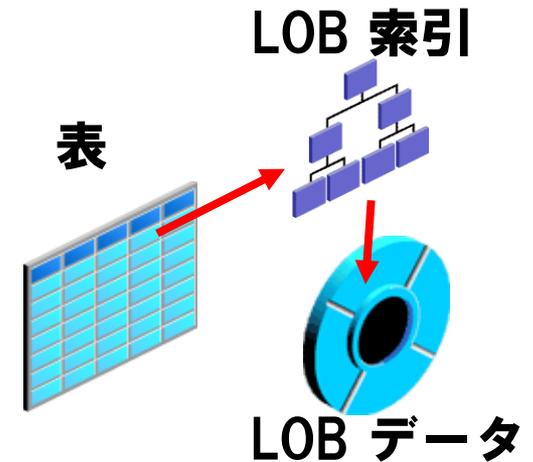
- チャンクはLOBデータを構成する要素
- LOBデータに対するI/Oはチャンク単位で行われます
 - BASICFILEのチャンクサイズは固定
 - SECUREFILEのチャンクサイズは可変で自動チューニング



LOB索引

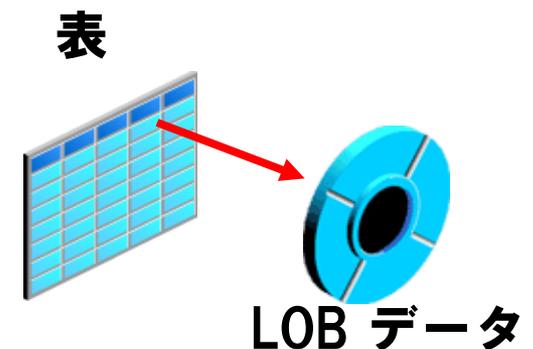
- BASICFILE

- アウトライン格納の場合に、LOB索引経由でLOBデータにアクセス
- LOB索引がLOBデータへのポインタを管理
- LOB索引で空き領域管理をおこなう
- LOB索引は内部的に作成、利用される



- SECUREFILE

- LOB索引は利用しない
(非重複機能利用時は除く)
- 表の各行でLOBデータへのポインタを管理
- 空き領域管理はメモリ(SGA)内にておこなう



LOBを含んだ表の作成 (BASICFILE)

```
CREATE TABLE lob_tab (  
  id          NUMBER(8) PRIMARY KEY,  
  blob_col BLOB)  
TABLESPACE USERS  
LOB(blob_col) STORE AS BASICFILE (  
  TABLESPACE LOBTBS  
  PCTVERSION 5  
  FREEPOOLS 1  
  CHUNK 32K  
  NOCACHE  
  LOGGING  
  DISABLE STORAGE IN ROW  
  STORAGE (  
    INITIAL 1M  
    NEXT 1M  
  )  
);
```

- **CHUNK**
 - LOBデータに対する実際のI/O単位(サイズ)を指定
 - 実際はdb_block_sizeの倍数に切り上げられる
- **FREEPOOLS**
 - LOBセグメントに対する空きリストのグループ数
 - 通常RACのインスタンス数
- **PCTVERSION**
 - LOBの記憶域全体のうち、旧バージョンのLOBの保持に使用される割合(%)の最大値指定
 - デフォルトは10

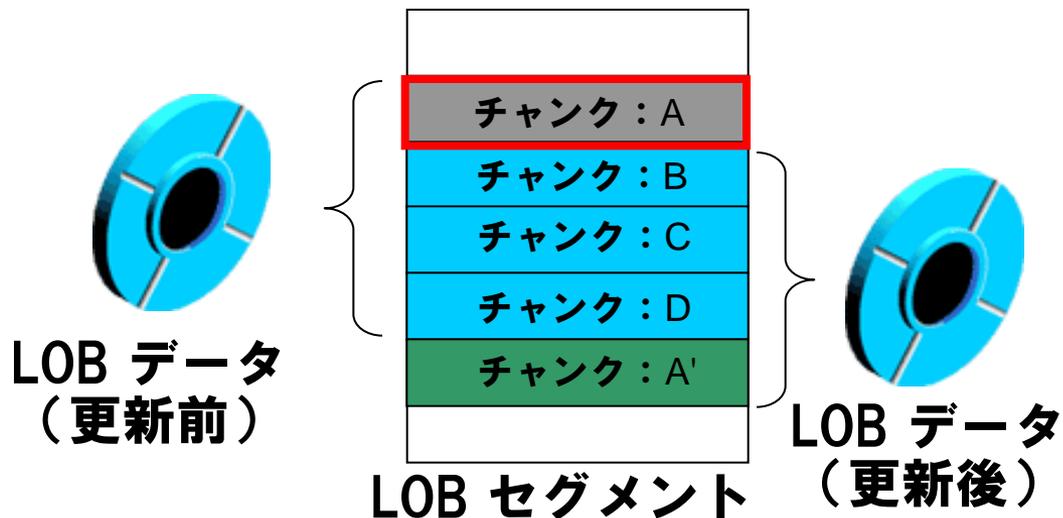
LOBを含んだ表の作成 (SECUREFILE)

```
CREATE TABLE lob_tab (  
  id          NUMBER(8) PRIMARY KEY,  
  blob_col   BLOB)  
TABLESPACE USERS  
LOB(blob_col) STORE AS SECUREFILE (  
  TABLESPACE LOBTBS  
  COMPRESS MEDIUM  
  ENCRYPT USING 'AES128'  
  DEDUPLICATE  
  NOCACHE  
  FILESYSTEM_LIKE_LOGGING  
  DISABLE STORAGE IN ROW  
  STORAGE (  
    INITIAL 1M  
    NEXT   1M  
  )  
);
```

- 以下のパラメータは設定できない (無視)
 - CHUNK
 - FREEPOOLS
 - PCTVERSION
- 圧縮、暗号化、非重複機能を利用可能 (EE Option機能)
 - COMPRESS
 - ENCRYPT
 - DEDUPLICATE
- メタデータのみREDOログに格納することが可能
 - FILESYSTEM_LIKE_LOGGING

LOBデータのUNDO（その1）

- UNDOはUNDOセグメントに格納されないことがある
 - LOBのインライン格納 → UNDOセグメントに格納
 - LOBのアウトライン格納 → LOBセグメントに格納
- コミットまでは上書きされないが、コミット後は上書き対象



- コミット後も変更前イメージを残すことで「読取り一貫性」を確保
- 上書きのタイミングの制御は以下のパラメータで決定
 - **PCTVERSION**
 - **RETENTION**

LOBデータのUNDO（その2）

- PCTVERSION（10.1までのデフォルト）
 - BASICFILEのみ利用可能
 - 使用中のLOBデータ領域全体のうち、旧バージョンのLOBデータ・ページが占めることのできる割合
 - 複数のトランザクションで更新がかかる場合は20以上に設定
 - 読み取り専用（更新しない）の場合は0に設定
- RETENTION（10.2からのデフォルト）
 - 自動UNDOモードの場合のみ設定することが可能
 - 自動セグメント領域管理（ASSM）の場合のみ有効（KROWN#130458）
 - 旧バージョンのLOBデータを保持する期間をUNDO_RETENTIONパラメータの期間（デフォルト 900秒）で制御
 - SECUREFILEの場合のみ、オプション指定可能
 - AUTO、MAX、MIN、NONE

LOBデータのキャッシュ

- Databaseのバッファ・キャッシュを利用するかどうかを選択可能（LOBのアウトライン格納の場合のみ）

CACHE :LOBデータがバッファ・キャッシュに配置される

NOCACHE :LOBデータはバッファ・キャッシュを利用しない
(LOBのデフォルト)

CACHE READS :WRITE操作ではバッファ・キャッシュは利用されず、
READ操作でバッファ・キャッシュに配置される

- **CACHE** オプションを使用すると、LOB列からのデータの読取りおよび書込み時のパフォーマンスが向上します。
- ただし、LOBでない他のページが通常より早くバッファ・キャッシュからページ・アウトされる可能性があります

※ セグメント記憶域に設定した場合と意味合いが異なるので注意

ORACLE

LOBデータのREDO

- LOB属性としてLOGGING以外を指定することで、LOBセグメントのREDO生成を抑制可能
 - LOBのアウトライン格納の場合のみ
 - CACHEオプションと同時に指定できない
 - BASICFILE: NOLOGGING
 - SECUREFILE: FILESYSTEM_LIKE_LOGGING
- 上記LOB属性は初期ローディングや、業務的にデータ復旧が可能な場合に利用
 - オンライン・バックアップ中にデータがディスクで変更される可能性がある為、オンライン・バックアップ取得前にはLOGGING属性に変更

http://download.oracle.com/docs/cd/E16338_01/appdev.112/b56263/adlob_lob_ops.htm#BACEDAEH

LOBデータのパーティショニング

```
CREATE TABLE lob_tab (  
  id          NUMBER(8) PRIMARY KEY,  
  blob_col   BLOB)  
LOB(blob_col) STORE AS SECUREFILE (  
  DISABLE STORAGE IN ROW  
  RETENTION AUTO  
  NOCACHE LOGGING  
)  
PARTITION BY RANGE(id) (  
  PARTITION p001  
    VALUES LESS THAN (100) TABLESPACE p001_tbs  
    LOB(blob_col) STORE AS SECUREFILE (TABLESPACE p001_lobtbs),  
  PARTITION p002  
    VALUES LESS THAN (200) TABLESPACE p002_tbs  
    LOB(blob_col) STORE AS SECUREFILE (TABLESPACE p002_lobtbs)  
) ;
```

← LOB全体の設定

↑ パーティション毎のLOBの設定

Agenda

- LOB概要
- Oracle Database のLOB
- LOBの管理
- ➔ • LOBの操作
- LOBを扱う上での留意点
- 事例



無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
 - MySQLからの移行相談
 - PostgreSQLからの移行相談
 - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
 - Oracle Developer/2000 Webアップグレード相談
 - パフォーマンス・クリニック
- Oracle Database 構成相談
- Oracle Database 高可用性診断
 - システム連携アセスメント
 - システムセキュリティ診断
 - 簡易業務診断
 - メインフレーム資産活用

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

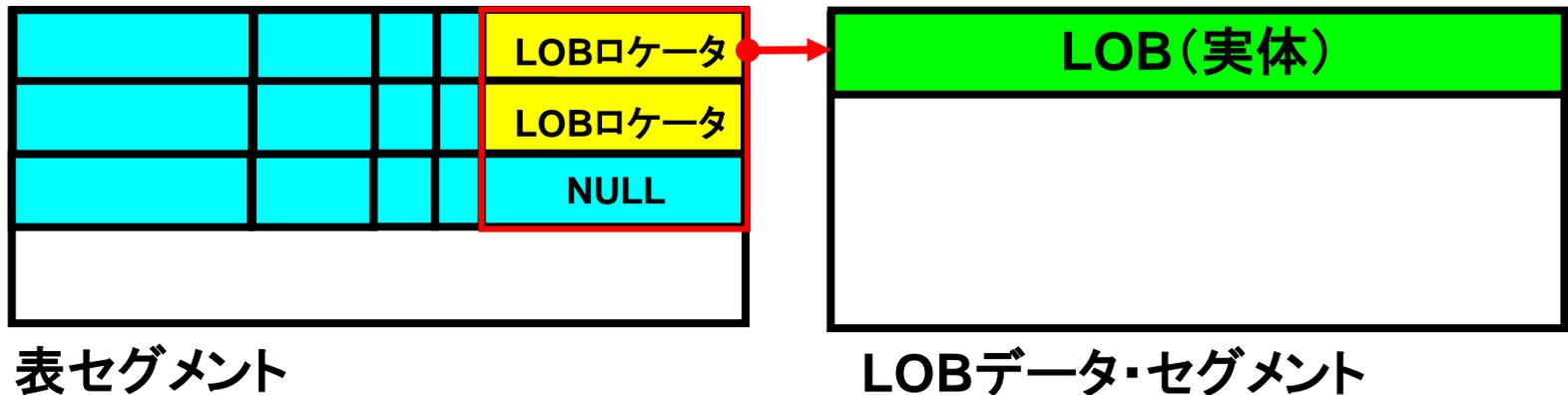
LOBとLOBロケータ

- 表のLOB列の状態は以下の2種類

NULL : LOBロケータがないNULLの状態

LOBロケータ有 : 長さ0のLOB (空のLOB)、もしくは実体が格納されたLOB

- LOBに対する操作にはLOBロケータが必要
 - NULLの場合は、UPDATEにて空のLOBを作成



LOBの操作（その1）

LOBロケータを利用したLOB値へのアクセスと変更

```
DECLARE
  CURSOR c1 IS SELECT id, blob_col FROM lob_tab;
  rec c1%ROWTYPE;
  buffer RAW(256);
  amount INTEGER;
BEGIN
  OPEN c1;
  LOOP
    FETCH c1 INTO rec;
    EXIT WHEN c1%NOTFOUND;
  BEGIN
    DBMS_LOB.OPEN
      (rec.blob_col, DBMS_LOB.LOB_READONLY);
    amount := DBMS_LOB.GETLENGTH(rec.blob_col);
    DBMS_LOB.READ
      (rec.blob_col, amount, 1, buffer);
    DBMS_OUTPUT.PUT_LINE(RAWTOHEX(buffer));
    DBMS_LOB.CLOSE(rec.blob_col);
  END;
  END LOOP;
  CLOSE c1;
END;
```

PL/SQLでBLOB
を読み込む例

※エラーハンドリングは省略しています

- LOBロケータを利用してAPI経由で、明示的にLOBを操作

```
#include <stdio.h>

int main(){
  FILE *fp;
  char dat[BUFSIZ];
  size_t size,i;

  fp = fopen("test.bin","r");
  size = fread(dat, 1, BUFSIZ, fp);
  for (i = 0; i <= size; i++) {
    printf("%x", dat[i]);
  }
  fclose(fp);
  exit(0);
}
```

C言語でFile
を読み込む例

※エラーハンドリングは省略しています

LOBの操作（その1:続き）

LOBロケータを利用したLOB値へのアクセスと変更

- LOBの初期化
 - 空のLOBを作成（LOBの実体は無し）

```
SQL> INSERT INTO lob_tab (id, blob_col) VALUES (2, EMPTY_BLOB());
```

```
SQL> INSERT INTO lob_tab (id, blob_col) VALUES (2, NULL);
```

```
SQL> UPDATE lob_tab SET blob_col = EMPTY_BLOB()
```

```
SQL> WHERE id = 2;
```

```
DECLARE
```

```
lob BLOB;
```

```
buffer RAW(256);
```

```
BEGIN
```

```
INSERT INTO lob_tab (id, blob_col) VALUES (3, empty_blob())
```

```
RETURNING blob_col INTO lob;
```

```
DBMS_LOB.OPEN(lob, DBMS_LOB.LOB_READWRITE);
```

```
buffer := HEXTORAW('0123456789ABCDEF');
```

```
DBMS_LOB.WRITE(lob, 8, 1, buffer);
```

```
DBMS_LOB.CLOSE(lob);
```

```
END;
```

INSERT実施後にLOB値をセットしている例

※エラーハンドリングは省略しています

LOBの操作 (その2)

LOB用データ・インターフェース

- 元々はLONGやLONG RAWを扱う為のAPI

```
DECLARE
  CURSOR c1 IS SELECT id, blob_col
                FROM lob_tab;
  TYPE rec_type IS RECORD (
    id lob_tab.id%TYPE,
    blob_col RAW(256)
  );
  rec rec_type;
BEGIN
  OPEN c1;
  LOOP
    FETCH c1 INTO rec;
    EXIT WHEN c1%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE(
      RAWTOHEX(rec.blob_col)
    );
  END LOOP;
  CLOSE c1;
END; ※エラーハンドリングは省略しています
```

PL/SQLでBLOB
を読み込む例

- アプリケーション側ではバインド変数型として、**LONG, LONG RAW** に該当するものを利用
 - Java API, PL/SQL API, OCI API を提供
- 利点
 - LOBロケータを意識しなくても良い
 - LONG型を利用している既存アプリケーションも、変更をほとんど加えずとも、DB側をLOBデータ型に変更して利用可能。
 - 最初にLOBロケータをFETCHしてから、LOB(実体)にアクセスする場合に比べると1つの動作で処理が完了
 - 配列のバインドおよび定義インターフェースを使用することで、LOBを含む複数の行を1回のラウンドトリップで処理可能
- いくつか制限があります(マニュアル参照)
 - ランダムアクセス不可
 - サーバ側内部ドライバで扱えるのは一度に4000Byteまで

ユーティリティでの操作 (SQL*Loader)

- LOB値を別ファイルから読み込むことができます

```
LOAD DATA
INFILE      test.dat
INTO TABLE lob_tab
TRUNCATE
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(
  id          integer external,
  ext_fname   FILLER CHAR(10),
  blob_col    lobfile(ext_fname) terminated by EOF
)
```

制御ファイル

```
1 disk.bmp
2 book.bmp
```

データ・ファイル

カレント・フォルダに存在すること

SQL Developer での操作

The screenshot shows the Oracle SQL Developer interface. The main window displays the 'LOB_TAB' table with two rows of data. The 'ID' column contains values 1 and 2, and the 'BLOB_COL' column contains BLOB data. A dialog box titled '値の編集' (Edit Value) is open, showing a 3D bar chart representing image data. The dialog box has tabs for '情報' (Info) and 'イメージ' (Image), and buttons for 'ロード(L)', '保存(S)', 'ヘルプ(H)', 'OK', and '取消' (Cancel). The '表示モード' (Display Mode) section has 'イメージ(I)' checked and 'テキスト(T)' unchecked.

ID	BLOB_COL
1	1 (BLOB)
2	2 (BLOB)

値の編集

イメージデータの表示可能

ロード(L) 保存(S) 表示モード: イメージ(I) テキスト(T)

ヘルプ(H) OK 取消

Agenda

- LOB概要
- Oracle Database のLOB
- LOBの管理
- LOBの操作
- ➔ • LOBを扱う上での留意点
- 事例



無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
 - MySQLからの移行相談
 - PostgreSQLからの移行相談
 - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
 - Oracle Developer/2000 Webアップグレード相談
 - パフォーマンス・クリニック
- Oracle Database 構成相談
- Oracle Database 高可用性診断
 - システム連携アセスメント
 - システムセキュリティ診断
 - 簡易業務診断
 - メインフレーム資産活用

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

LOBの管理上の留意点

1. 頻繁に領域拡張が発生しないように設計する
 - EXTENTサイズを十分に大きくする（例:64MB）
2. 頻繁にREDOログがスイッチしないように
 - REDOログサイズを十分に大きくする
3. 再実行可能な情報（セッション情報等）を格納する場合はNOLOGGING/FILESYSTEM_LIKE_LOGGINGの利用を検討する
 - REDOログ生成量の削減
4. INSERT競合が発生する場合
 - パーティション化を検討

LOBのパフォーマンス留意点 (BASICFILE)

1. CHUNKサイズを適切に(アウトラインLOBの場合のみ)
 - CHUNKのデフォルト・サイズは表領域ブロック・サイズと同じ
 - 最大値は32KB
 - LOB列の作成時にCHUNKの値を選択した後は、値を変更できない
2. サイズの小さなLOB(8KB以下)を扱う場合
 - インラインLOB(ENABLE STORAGE IN ROW)を利用
 - ブロックサイズを8KBにし、CHUNKサイズも8KBに

Agenda

- LOB概要
- Oracle Database のLOB
- LOBの管理
- LOBの操作
- LOBを扱う上での留意点
- 事例



無償技術サービスOracle Direct Concierge

- SQL Serverからの移行アセスメント
 - MySQLからの移行相談
 - PostgreSQLからの移行相談
 - Accessからの移行アセスメント
- Oracle Database バージョンアップ支援
 - Oracle Developer/2000 Webアップグレード相談
 - パフォーマンス・クリニック
- Oracle Database 構成相談
- Oracle Database 高可用性診断
 - システム連携アセスメント
 - システムセキュリティ診断
 - 簡易業務診断
 - メインフレーム資産活用

<http://www.oracle.com/lang/jp/direct/services.html>

ORACLE

事例

- National Ignition Facility (SecureFiles)

http://www.oracle.com/technology/products/database/securefiles/pdf/securefiles_nif.pdf

- Melbourne Health (DICOM ※)

<http://www.oracle.com/customers/snapshots/melbourne-health-and-biogrid-australia-case-study.pdf>

- A Performance Evaluation of Storage and Retrieval of DICOM Content in Oracle Database 11g Using HP Blade Servers and Intel Processors

http://www.oracle.com/technology/products/intermedia/pdf/ora_dicom_bench_2008.pdf

※ DICOMは医療画像情報を保存する為のオブジェクト型で、内部的にBLOBを利用しています

ORACLE

まとめ

- すべてのデータをOracle Databaseで一元管理することにはいろいろなメリットがあります
 - アプリケーション・ロジックが単純化できる
 - 一つのトランザクションで操作が可能に
 - バックアップなどの運用を一元化できる
- LOBは通常の実データとは若干異なりますが、動作や扱いが理解できると、それほど難しいものではありません
- LOBはサイズが大きなデータを扱うことが多い為、そのあたりを考慮した設計が必要となります

OTN×ダイセミ でスキルアップ!!



- ・技術的な内容について疑問点を解消したい！
- ・一般的なその解決方法などを知りたい！
- ・ 세미나資料など技術コンテンツがほしい！

Oracle Technology Network(OTN)を御活用下さい。

<http://otn.oracle.co.jp/forum/index.jspa?categoryID=2>

技術的な疑問点は、OTN揭示版の
「データベース一般」へ

※OTN揭示版は、基本的にOracleユーザー有志からの回答となるため100%回答があるとは限りません。
ただ、過去の履歴を見ると、質問の大多数に関してなんらかの回答が書き込まれております。

<http://www.oracle.com/technology/global/jp/ondemand/otn-seminar/index.html>

過去のセミナー資料、動画コンテンツはOTNの
「OTNセミナー オンデマンドコンテンツ」へ

※ダイセミ事務局にダイセミ資料を請求頂いても、お受けできない可能性がございますので予めご了承ください。
ダイセミ資料はOTNコンテンツ オン デマンドか、セミナー実施時間内にダウンロード頂くようお願い致します。

ORACLE

OTNセミナー オンデマンド コンテンツ

期間限定にて、ダイセミの人気セミナーを動画配信中!!

ダイセミのライブ感はそのままに、お好きな時間で受講頂けます。

最新のコンテンツ

エンジニアのためのITIL実践術 再生時間: 60分	ここからはじめよう Oracle PL/SQL入門 再生時間: 60分	実践!!高可用システム構築 -RAC基本 再生時間: 60分	お悩み解決! Oracle のサイジング 再生時間: 60分

Database

今さら聞けない!!バックアップ-リカバリ入 再生時間: 60分	意外と簡単!?! Oracle Database 11g -セ 再生時間: 60分	実践!!バックアップ-リカバリ 再生時間: 60分	意外と簡単!?! Oracle Database 11g -デ 再生時間: 60分

>> もっと見る

OTN オンデマンド

検索

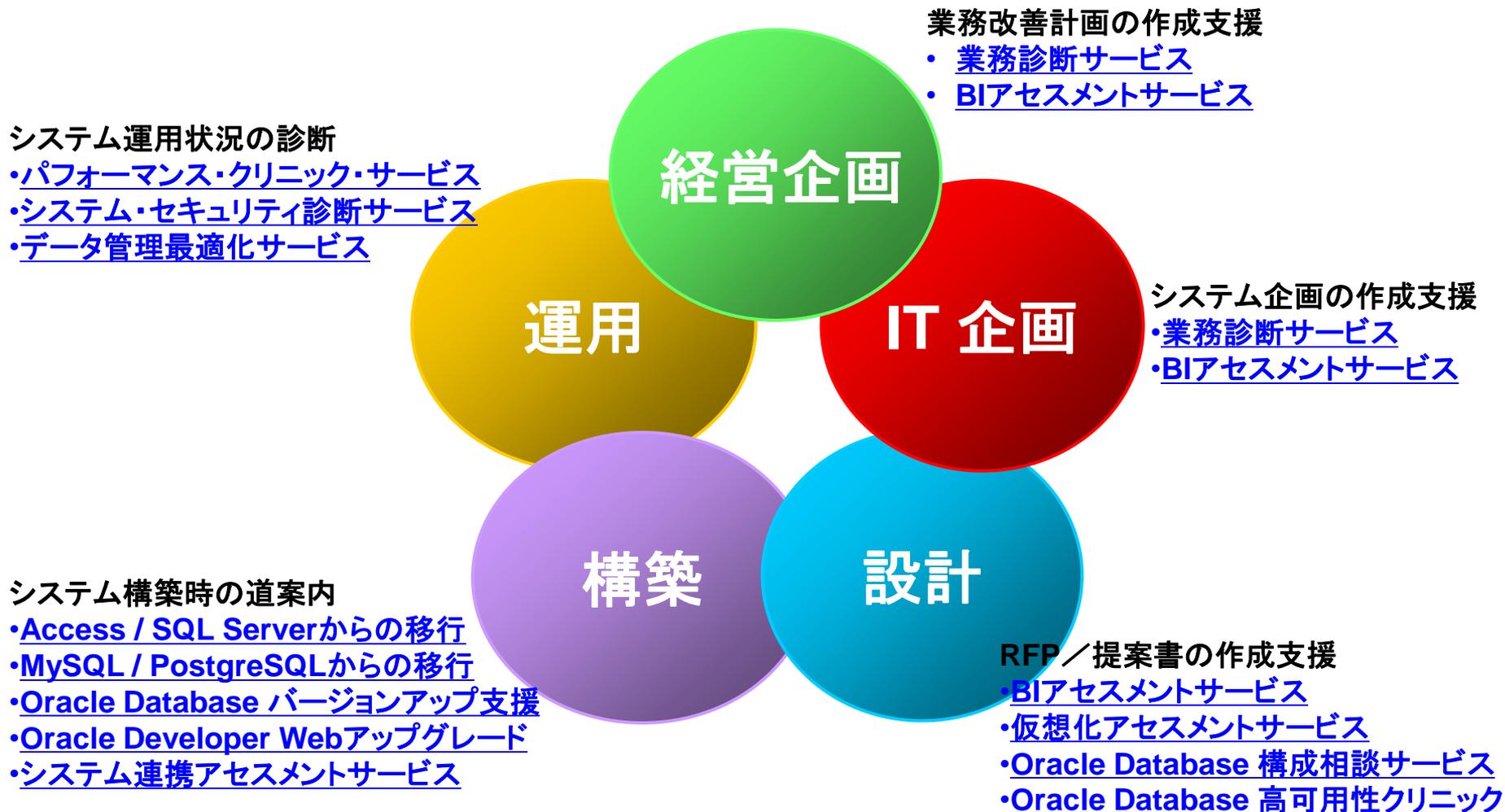
※掲載のコンテンツ内容は予告なく変更になる可能性があります。

期間限定での配信コンテンツも含まれております。お早めにダウンロード頂くことをお勧めいたします。

ORACLE

ITプロジェクト全般に渡る無償支援サービス

Oracle Direct Conciergeサービスメニュー



あなたにいちばん近いオラクル



Oracle Direct

まずはお問合せください

システムの検討・構築から運用まで、ITプロジェクト全般の相談窓口としてご支援いたします。

システム構成やライセンス/購入方法などお気軽にお問い合わせ下さい。

Web問い合わせフォーム

専用お問い合わせフォームにてご相談内容を承ります。

http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28

※フォームの入力には、Oracle Direct Seminar申込時と同じ
ログインが必要となります。

※こちらから詳細確認のお電話を差し上げる場合がありますので、ご登録されている連絡先が最新のものになっているか、ご確認下さい。

フリーダイヤル

0120-155-096

※月曜～金曜 9:00～12:00、13:00～18:00

(祝日および年末年始除く)

ORACLE®



以上の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して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、弊社の裁量により決定されます。

Oracle、PeopleSoft、JD Edwards、及びSiebellは、米国オラクル・コーポレーション及びその子会社、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の可能性がります。