

Oracle Database Technology Night ～ 集え！オラクルの力(チカラ) ～

18^c ORACLE[®]
Database

Oracle Database 18c テクノロジーシリーズ 4 「Development と Performance 関連の機能強化」 ～ Development ～

日本オラクル株式会社
ソリューション・エンジニアリング統括
クラウド・インフラストラクチャー本部
黒瀬 統洋

ORACLE[®]

- 以下の事項は、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とJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

はじめに: 「新機能」というよりは、「改良」

各種標準への準拠をコツコツと

大量データの分析に役立つ機能

外部・非構造化データへの対応 (JSON含)

メンテナンス性向上 => Autonomousへの道?

より動的な定義を可能に => Autonomousへの道?

Oracle Database 18c – SQL改良点、本日のハイライト

18c

最近接偶数丸め

近似上位N問合せ処理

多相表(Polymorphic Table Functions)

JSON

メンテナンス性向上関連

ROUND_TIES_TO_EVEN() ファンクション

- 構文

ROUND_TIES_TO_EVEN(*n* [, *integer*])

- 対象桁が中間値の時の動作がROUNDと異なる
 - 端数が中間値(0.5など)の時切り捨てと切り上げのうち結果が偶数となる方へ丸める
 - 「最近接偶数丸め」「偶数丸め」「JIS丸め」「ISO丸め」「五捨五入」「偶捨奇入」「bankers' rounding」
- JIS Z 8401の規則A、ISO 80000-1
- BINARY_FLOAT と BINARY_DOUBLE は対象外

ROUND() と ROUND_TIES_TO_EVEN() の比較

Value	ROUND (Value, 0)	ROUND_TIES_TO_EVEN (Value, 0)
0.4	0	0
0.5	1	0
0.6	1	1
1.5	2	2
2.5	3	2
3.5	4	4
4.5	5	4

端数が中間値(0.5など)の時切り捨てと切り上げのうち結果が偶数となる方へ丸める

SUM()ってみると...

Value	ROUND (Value, 0)	ROUND_TIES_TO_EVEN (Value, 0)
0.4	0	0
0.5	1	0
0.6	1	1
1.5	2	2
2.5	3	2
3.5	4	4
4.5	5	4

sum(value)=**13.5**

sum(round(value,0))=**16**

誤差の積み重ねによる偏りが少ない

Value	ROUND (Value, 0)	ROUND_TIES_TO_EVEN (Value, 0)
0.4	0	0
0.5	1	0
0.6	1	1
1.5	2	2
2.5	3	2
3.5	4	4
4.5	5	4

sum(value)=**13.5**

sum(round(value,0))=**16**

sum(round_ties_to_even(value,0))=**13**

Oracle Database 18c – SQL改良点、本日のハイライト

最近接偶数丸め

18c

近似上位N問合せ処理

多相表(Polymorphic Table Functions)

JSON

メンテナンス性向上関連

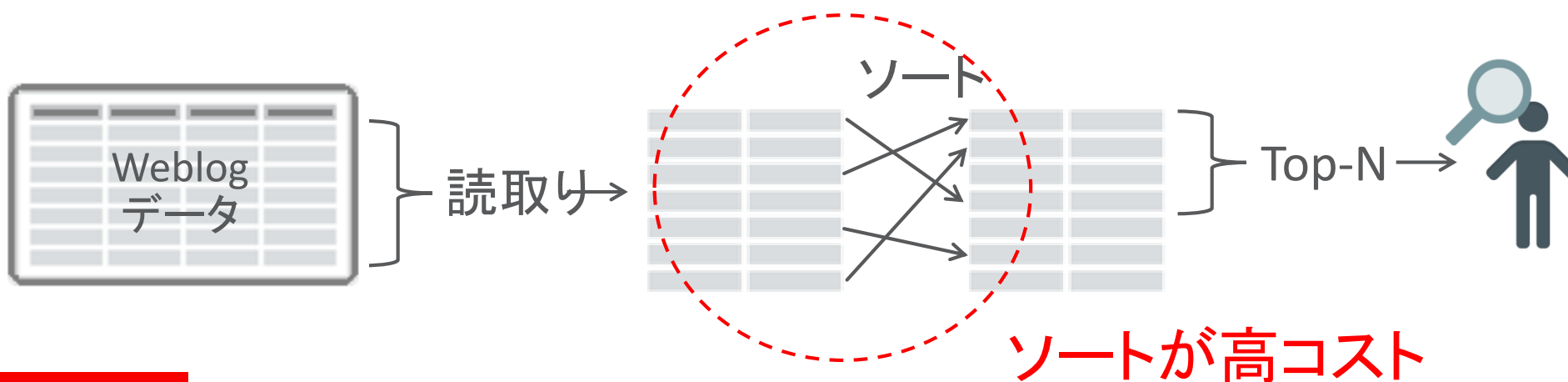
TOP-N問合せ(ANSI SQL:2008)

11gまで
Window関数row_number()を使用し、
Order by をかける

```
SELECT empno, ename, deptno  
FROM (SELECT empno, ename, deptno, sal, comm,  
            row_number() OVER (ORDER BY sal,comm) rn  
      FROM emp  
      )  
WHERE rn <=5  
ORDER BY sal , comm;
```

12c以降
TOP-N文をサポート
ANSI標準に沿ったシンプル構文

```
SELECT empno, ename, deptno  
FROM emp  
ORDER BY sal, comm FETCH FIRST 5 ROWS ONLY;
```



近似問合せ

- PERCENTILE CONT, PERCENTILE DISC, MEDIAN
 - ソートを必要とし、大きなリソースを要求する従来の関数
- 近似関数の導入:
 - APPROX_PERCENTILE
 - APPROX_MEDIAN
 - APPROX_xxxx_DETAIL, APPROX_xxxx_AGG
 - TO_APPROX_xxxx
- エラー範囲 約 **0.1 - 1.0%**
- **6-13倍のパフォーマンス向上 (TPC-H)**

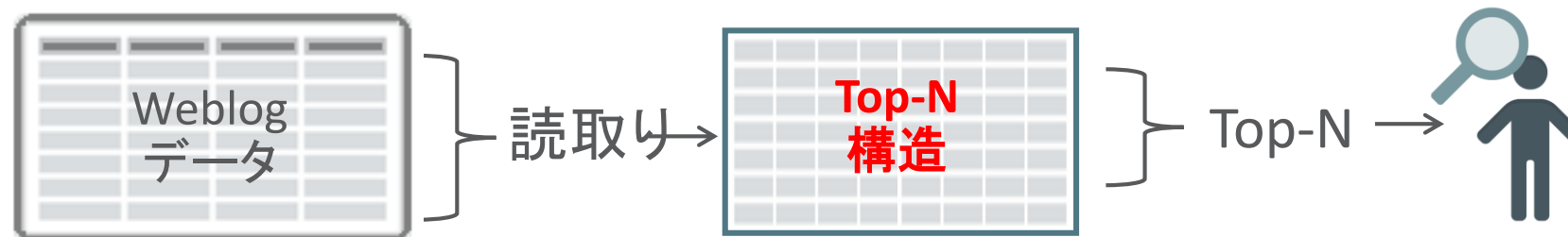
APPROX_PERCENTILEの利点: 13倍 高速

Operation	Name	Lin...	Estimated ...	Cost	Timeline(187s)	Execu...	Actual R...	Memory (...)	Temp (Max)	O...
SELECT STATEMENT		0				1	1			
SORT AGGREGATE		1	1			1	1			
PX COORDINATOR		2				65	32			
PX SEND QC (RANDOM)	:TQ10001	3	1			32	32			
SORT AGGREGATE		4	1			32	32			
VIEW		5	15	5,084		32	15			
SORT GROUP BY		6	15	5,084		32	15	1GB	11GB	

Operation	Name	Lin...	Estimated ...	Cost	Timeline(14s)	Execu...	Actual R...	Memory (...)	Temp (Max)	O...
SELECT STATEMENT		0				1	1			
SORT AGGREGATE		1	1			1	1			
PX COORDINATOR		2				65	32			
PX SEND QC (RANDOM)	:TQ10001	3	1			32	32			
SORT AGGREGATE		4	1			32	32			
VIEW		5	15	5,084		32	15			
SORT GROUP BY APPROX		6	15	5,084		32	15	830KB		

Top-N 問合せ x 近似関数

- 近似関数の拡充:
 - APPROX_COUNT() APPROX_RANK() APPROX_SUM()
- 高パフォーマンス
 - 大きなデータセットに対して著しい効果
- 高い正確性
- “Top-N 構造” はメモリ上の小さい構造
 - ディスクソート無し



Top-N approximate aggregation

テラバイトのデータに対するインタラクティブな応答時間

- **TOP-N 問合せに対する概算結果**

- おおよそどの程度のページ・ビューが上位 5 つの blog 投稿に対して先週あったか?
- 各地域のトップ 50 の顧客と、そのおおよその支出は?

近似ヒット数によるトップ 5 blogs

```
SELECT blog_post, APPROX_COUNT(*)  
FROM weblog  
GROUP BY blog_post  
FETCH FIRST 5 ROWS ONLY;
```

各地域のトップ 50 顧客とおおよその支出

```
SELECT region, customer_name,  
       APPROX_RANK(PARTITION BY region  
                   ORDER BY APPROX_SUM(sales) DESC) appr_rank,  
       APPROX_SUM(sales) appr_sales  
FROM sales_transactions  
GROUP BY region, customer_name  
HAVING APPROX_RANK(..) <=50;
```

Oracle Database 18c – SQL改良点、本日のハイライト

最近接偶数丸め

近似上位N問合せ処理

18c

多相表(Polymorphic Table Functions)

JSON

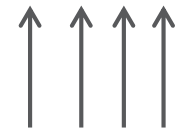
メンテナンス性向上関連

Polymorphicテーブル: 自己記述式 ダイナミックSQL



SQL
問合せ

STATE_ID	POP	LOANS	A_LOAN	A_SCORE	RISK
					H
					H
					H
					H



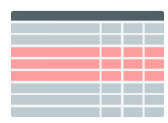
POLYMORPHIC TABLE FUNCTION

ブラックボックス化

クレジットリスク
アルゴリズム



入力:



TABLE



XML



JSON



ANALYTIC VIEW

- ANSI SQL:2016
- FROM句で使用可能なユーザ定義関数
- 高度なアルゴリズムを隠蔽
 - アルゴリズムを隠蔽化
 - SQLの柔軟性
 - どんなテーブル列も処理可能
 - SQL rowset (table, JSON, XML doc など)を返す
 - 例) クレジットスコアとリスクレベルを返す
そのアルゴリズムはブラックボックス化



Polymorphicテーブル: 使用例1

クレジットリスク 評価モデルの隠蔽化



SQL
問合せ

STATE_ID	POP	LOANS	A_LOAN	A_SCORE	RISK
					H
					H
					H
					H



```
SELECT
  state_id,
  . . . ,
  AVG(credit_score),
  risk
FROM CREDIT_RISK (
  tab => scott.customers,
  cols => columns(dob, zip,
                  loan_default),
  outs => columns(credit_score,
                  risk_level))
WHERE risk_level = 'High'
GROUP BY state_id;
```

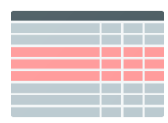
POLYMORPHIC TABLE FUNCTION



ブラックボックス化

クレジットリスク
アルゴリズム

入力:



TABLE



XML



JSON



ANALYTIC VIEW



Polymorphicテーブル: 使用例2

- to_doc() PTF をつくり、表をJSONドキュメント形式に変換してみる
<https://livesql.oracle.com/apex/livesql/docs/Inpls/ptf/ptftodoc.html>

```
SELECT deptno, ename, document
FROM to_doc(scott.emp,
columns(empno,job,mgr,hiredate,sal,comm))
WHERE deptno IN (10, 30)
ORDER BY 1, 2;
```

DEPTNO	ENAME	DOCUMENT
10	CLARK	{"EMPNO":7782, "JOB":"MANAGER", "MGR":7839, "HIREDATE":"09-JUN-81", "SAL":2450}
10	KING	{"EMPNO":7839, "JOB":"PRESIDENT", "HIREDATE":"17-NOV-81", "SAL":5000}
10	MILLER	{"EMPNO":7934, "JOB":"CLERK", "MGR":7782, "HIREDATE":"23-JAN-82", "SAL":1300}
30	ALLEN	{"EMPNO":7499, "JOB":"SALESMAN", "MGR":7698, "HIREDATE":"20-FEB-81", "SAL":1600, "COMM":300}
30	BLAKE	{"EMPNO":7698, "JOB":"MANAGER", "MGR":7839, "HIREDATE":"01-MAY-81", "SAL":2850}
30	JAMES	{"EMPNO":7900, "JOB":"CLERK", "MGR":7698, "HIREDATE":"03-DEC-81", "SAL":950}
30	MARTIN	{"EMPNO":7654, "JOB":"SALESMAN", "MGR":7698, "HIREDATE":"28-SEP-81", "SAL":1250, "COMM":1400}
30	TURNER	{"EMPNO":7844, "JOB":"SALESMAN", "MGR":7698, "HIREDATE":"08-SEP-81", "SAL":1500, "COMM":0}
30	WARD	{"EMPNO":7521, "JOB":"SALESMAN", "MGR":7698, "HIREDATE":"22-FEB-81", "SAL":1250, "COMM":500}

パラレル実行、一時表、Result Cache利用などもそのまま利用可能

Oracle Database 18c – SQL改良点、本日のハイライト

最近接偶数丸め

近似上位N問合せ処理

多相表(Polymorphic Table Functions)

 18c JSON

メンテナンス性向上関連

Oracle Database 12c “Under the Radar” Features

Security Assessment Tool

JSON

Application Continuity

Longer Varchars

SQL Plan Management Enhancements

Real Time Materialized Views

Auto Generated Sequences

Auto List Partitioning

Online Tablespace Encryption

Long Identifiers

Invisible Columns

Property Graph

Live SQL

Top N Queries

Online Table Move

Index Compression

Index Usage Stats

SQLによるJSONデータの問い合わせ

- 単純な問い合わせ

```
select j.PO_DOCUMENT  
from J_PURCHASEORDER j  
where j.PO_DOCUMENT.PONumber = 1600
```

- JSONパス式を使用した高度な問い合わせ

```
select JSON_VALUE(PO_DOCUMENT, '$.LineItems[0].Part.UnitPrice' returning NUMBER(5,3))  
from J_PURCHASEORDER p  
where JSON_VALUE(PO_DOCUMENT, '$.PONumber' returning NUMBER(10)) = 1600
```

– SQL2016として提案された文法に準拠

18cでの、JSONの主な改良

Simpler development of JSON-centric applications using Oracle Database

- TREAT (<expression> AS JSON) operator
 - JSONとして扱うオペランドを動的に宣言可能
- 大きなJSONドキュメントを生成可能
 - LOBをサポート
- JSON search indexesのキーを長く
 - 64 バイトから255 バイトへ; 長いキーを持つJSONオブジェクトの検索が速く
- 新しい SODA (Simple Oracle Document Access) ドライバ
 - 従来のJava と REST に加え、OCI と PL/SQL用SODA
 - JSON dataへのシンプルな、非リレーショナル ('nosql-like') APIアクセス
 - Node-oracledb、cx_Oracle、ODPI-C もSODAに対応 (new!)

TREAT (<expression> AS JSON) operator

JSONとして扱うオペランドを動的に宣言可能

```
drop table documents;  
CREATE TABLE documents  
  (id number NOT NULL,  
   data CLOB --, CONSTRAINT json_check CHECK (data IS JSON)  
  );  
INSERT INTO documents VALUES (1, '{"name":"apple", "quantity":10}');  
  
WITH json_docs AS  
  (SELECT id,  
         TREAT(data AS JSON) AS json_data  
   FROM documents)  
SELECT jd.id,  
       jd.json_data.name,  
       jd.json_data.quantity  
FROM   json_docs jd  
ORDER BY 1;
```

SODA for PL/SQLの例

JSONデータへのシンプルな、非リレーショナル ('NoSQL-like') APIアクセス

```
DECLARE
```

```
  collection SODA_COLLECTION_T;  
  document SODA_DOCUMENT_T;
```

```
BEGIN
```

```
- Open the collection
```

```
collection := DBMS_SODA.open_collection('myCollectionName');
```

```
- Find a document using a key
```

```
document := collection.find_one('key1');
```

```
IF document IS NOT NULL THEN
```

```
  DBMS_OUTPUT.put_line('Key: ' || document.get_key);
```

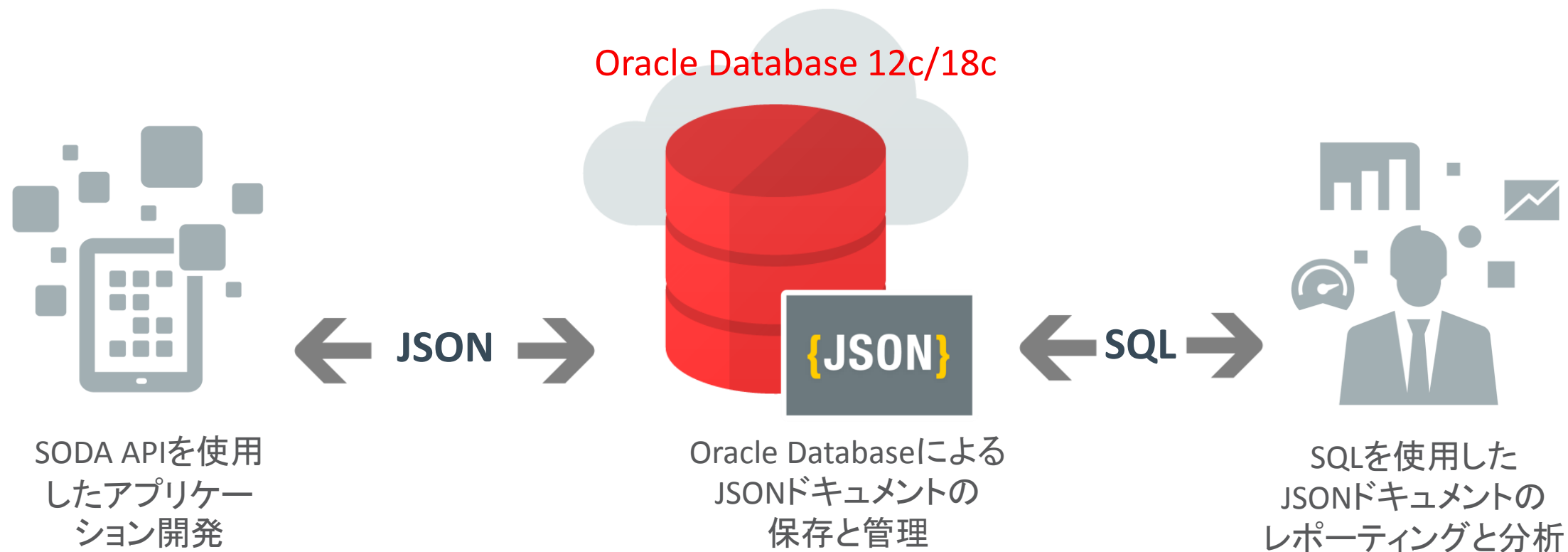
```
  DBMS_OUTPUT.put_line('Content: ' || utl_raw.cast_to_varchar2(document.get_blob));
```

```
END IF;
```

```
END;
```

```
/
```


Oracle 12c/18c JSONドキュメント・ストア エンタープライズ・クラスのデータ管理



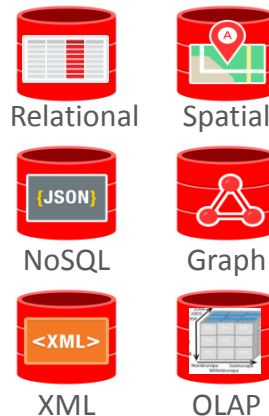
Polyglot Persistenceに対するオラクル製品戦略

どちらもサポート – 開発者に自由な選択を

シングルモデル

- オラクルはシングルモデルのデータストアを複数サポート

- リレーショナル
- キー/バリュー
- XML
- スペーシャル
- グラフ
- OLAP



- オラクルはシングルモデルのpolyglot環境をBig Data SQLで統合

マルチモデル

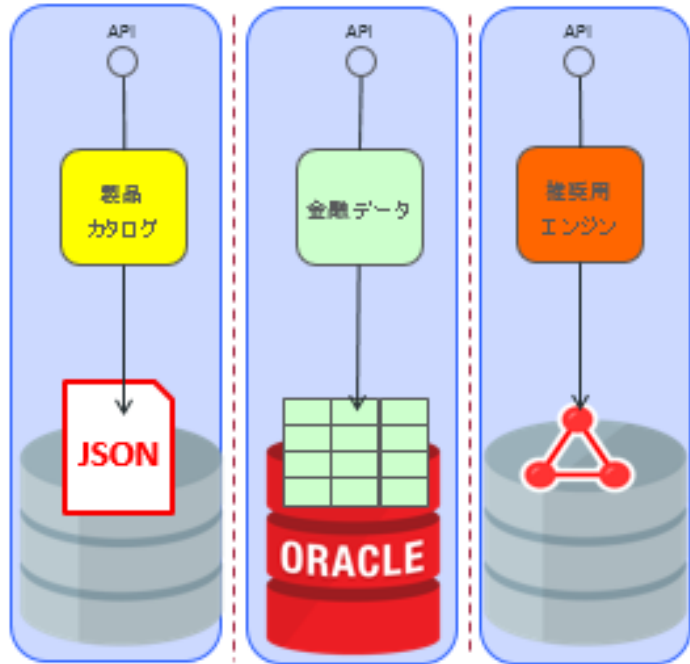
- Oracleデータベースは複数のデータオブジェクトをサポート

- リレーショナル
- JSON
- グラフ & スペーシャル
- XML
- Text

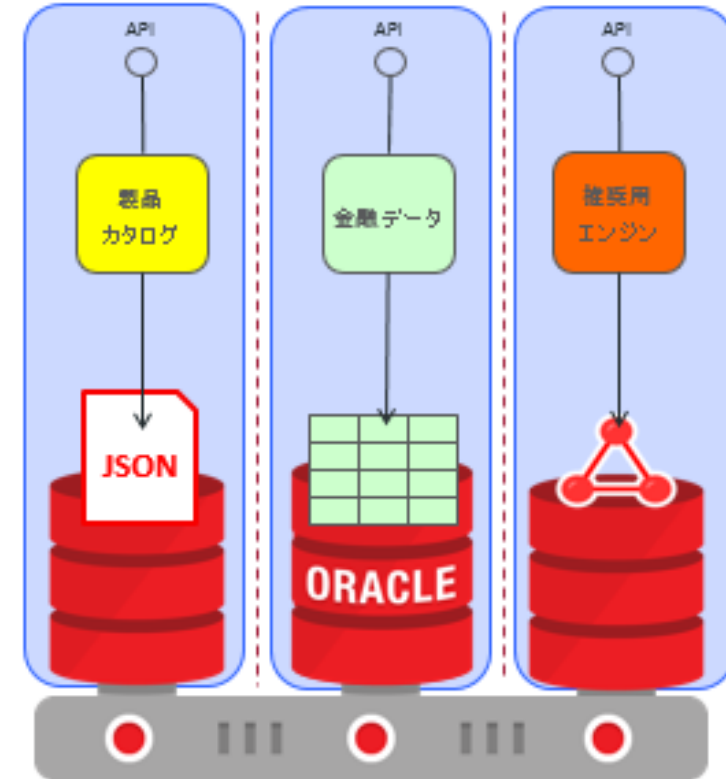


- Oracleデータベースは、全データベースオブジェクトへの統合アクセスを提供

マイクロサービス と Polyglot Persistence



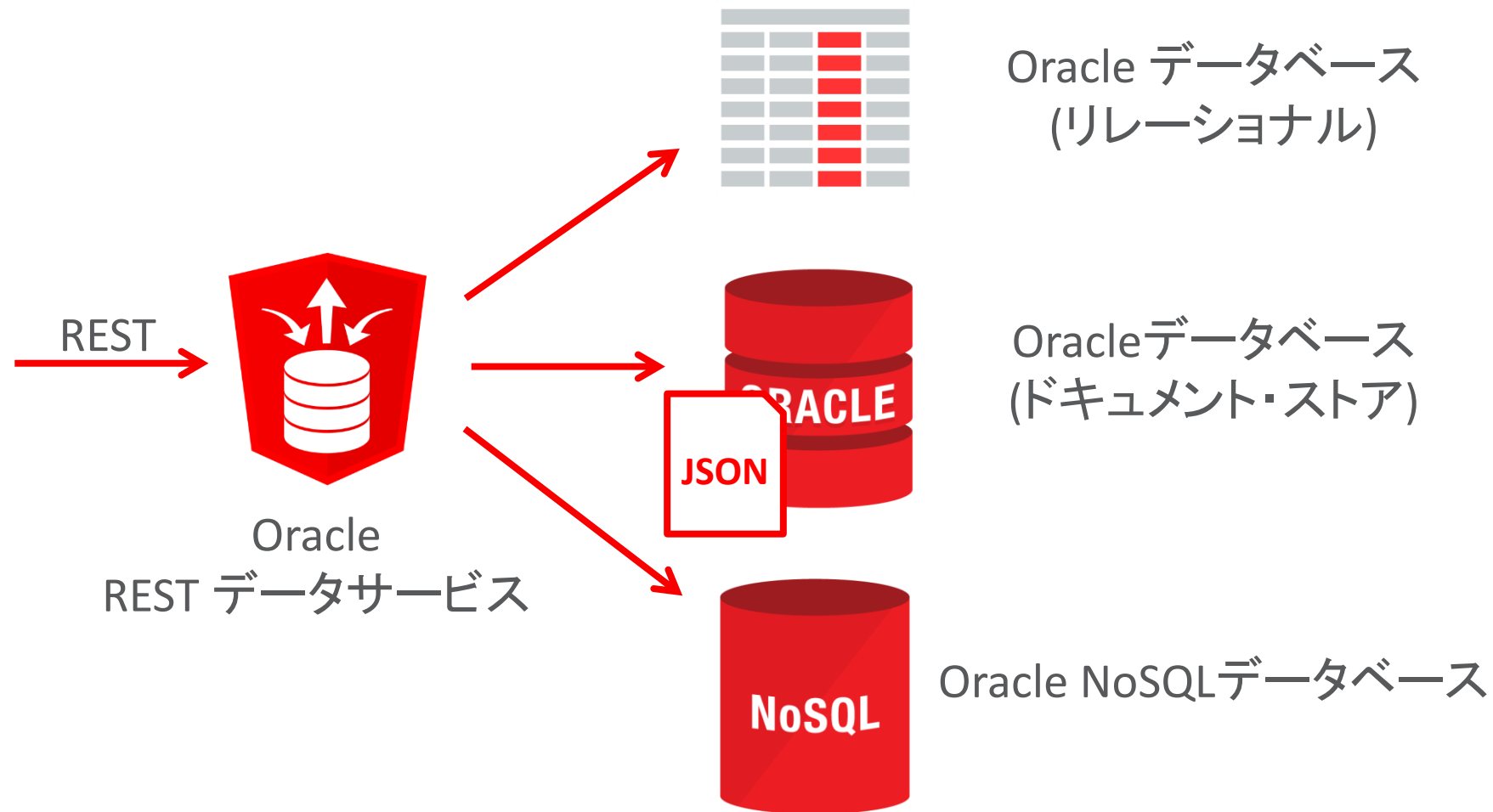
- **シングルモデルによる特殊化** の利点
 - 用途特化された API
 - 用途特化されたデータフォーマット
 - 用途特化されたアクセス法とインデックス



- **マルチモデルによる統合と標準化** の利点
 - 管理の標準化
 - 統合されたデータセキュリティポリシー
 - 複数のデータ型をまたがったシンプルインテグレーション
 - トランザクションとデータ一貫性

Oracle REST データサービス

REST-enable your data



Oracle Database 18c – SQL改良点、本日のハイライト

最近接偶数丸め

近似上位N問合せ処理

多相表(Polymorphic Table Functions)

JSON



メンテナンス性向上関連

Timeout / Termination

実行中SQLの手動終了

```
ALTER SYSTEM CANCEL SQL 'SID, SERIAL, @INST_ID, SQL_ID';
```

DBAがロングランニングな高コストSQLを、セッションキルせず終了可能

パラレル文のタイムアウト

```
PARALLEL_QUEUE_TIMEOUT
```

CANCEL → ORA-07454

RUN – 低いパラレル度での実行へダウングレードされる可能性はある

OCI Call Timeout Attribute

```
OCIAttrSet(... (ub4) OCI_ATTR_CALL_TIMEOUT ...)
```

```
OCIAttrGet(... (dvoid *) &get_timeout ... (ub4) OCI_ATTR_CALL_TIMEOUT ... )
```

ORA-03136 inbound connection timed out

Data Pump スキップ継続ロード ロードエラー発生時、スキップ可能に

CONTINUE_LOAD_ON_FORMAT_ERROR パラメータ (impdpの DATA_OPTIONS)

データロード中にストリームフォーマットエラーで失敗した場合、スキップ実行を続ける選択が可能
ダンプファイルが破損しており、元のダンプファイルが入手不可の場合など

現在の動作では、データ・ストリームのいずれかの部分に破損があると、Data Pumpが表データをロードできなくなる。この新しいオプションを使用すると、少なくとも読取り可能なデータはインポート可能に。

```
> impdp scott DUMPFILE=dpump_dir:emp.dmp DATA_OPTIONS=CONTINUE_LOAD_ON_FORMAT_ERROR
```

DBMS_DATAPUMP.SET_PARAMETER の DATA_OPTIONS にて KU\$_DATAOPT_CONT_LOAD_ON_FMT_ERR を指定することも出来る

UTL_I18N 関数: キャラクタ データ 検査

VARCHAR2, NVARCHAR2, CLOB, NCLOB のinvalid キャラクタを検出するファンクション
アプリが不正データを扱っていないことの確認が可能

DBキャラクタセット もしくは NCHARキャラクタセットに対する不正なコードポイントのオフセットを返す

```
CREATE TABLE temp(col1 NVARCHAR2(20), col2 CLOB);
INSERT INTO temp VALUES(UNISTR('foo¥D800bar'), UNISTR('foo¥D800bar'));

SELECT
  UTL_I18N.VALIDATE_CHARACTER_ENCODING(col1) invalid_offset_column1,
  UTL_I18N.VALIDATE_CHARACTER_ENCODING(col2) invalid_offset_column2
FROM temp;
```

INVALID_OFFSET_COLUMN1

INVALID_OFFSET_COLUMN2

7

4

(nchar = AL16UTF16)

(DB char = AL32UTF8)

オンライン 表 メンテナンス操作

- Oracle Database 11g
 - Create index
 - Add column with default value
 - Add constraint
- Oracle Database 12cリリース1
 - Drop Index
 - Drop Constraint
 - Alter table set unused column
 - Alter table move partition
- Oracle Database 12cリリース2
 - Alter table move for heap tables
 - Alter table split partition
 - Alter table modify non-partitioned table to partitioned table
- **New in 18c**
 - **Alter table merge partition**
 - **Alter table modify partitioned table to partitioned table**

参考: alter table merge partition ... online

パーティションをオンライン中に自由に変更



```
alter table sales merge partitions p3,  
p4 into partition p3a update indexes  
online;
```

https://livesql.oracle.com/apex/livesql/file/content_GAD3PVUACH5QQJYV7KJXB0POV.html

<https://connor-mcdonald.com/2018/02/27/18c-merge-partition-online/>

SPA Result Set Validation

SPAで結果セットの比較を行い、異なっているかレポート

結果セットをハッシュ化、試行1と試行2の結果が異なっているか確認

試行1と試行2でデータセットが異なっていると正しいテストとならない (SPAあるある)

- テストが正しい条件で行われているかを確認することが出来る
- 同じデータセット、異なる結果セットの場合は精査

SET_ANALYSIS_TASK_PARAMETER プロシージャで COMPARE_RESULTSETを使う

```
EXEC DBMS_SQLPA.SET_ANALYSIS_TASK_PARAMETER(task_name => 'my_spa_task', -  
parameter => 'COMPARE_RESULTSET', value => 'TRUE');
```

TEST EXECUTEで使用可

再掲：「新機能」というよりは、「改良」

各種標準への準拠をコツコツと

大量データの分析に役立つ機能

外部・非構造化データへの対応 (JSON含)

メンテナンス性向上 => Autonomousへの道？

より動的な定義を可能に => Autonomousへの道？

Appendix: その他、SQL改良点

プライベート一時表

インライン外部表

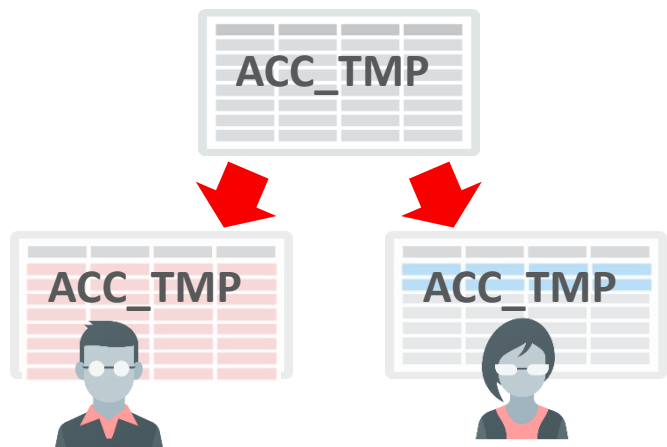
列ベース照合

分析ビュー

-- More -- (80%)

プライベート一時表

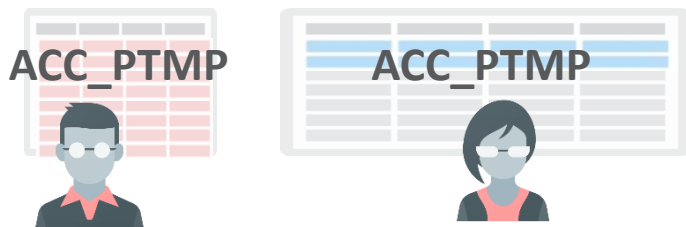
レポート向けアプリなどに活用できる一時表



グローバル一時表

- 永続する、共有された (global) 表定義
- 一時的な、プライベート (session-based) データコンテンツ
 - セッション あるいは トランザクション中、データは物理的に存在
 - セッションプライベートな統計

プライベート一時表 (New in 18c)



- 一時的な、プライベート (session-based) 表定義
 - プライベートな表名と定義
- 一時的な、プライベート (session-based) データコンテンツ
 - セッション あるいは トランザクション中

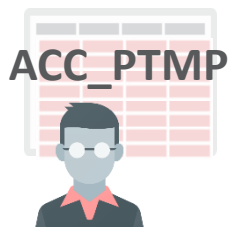
プライベート一時表

レポート向けアプリなどに活用できる一時表

```
CREATE PRIVATE TEMPORARY TABLE ORA$PTT_sales_ptt_transaction  
(time_id DATE,  
 amount_sold NUMBER(10,2))  
ON COMMIT DROP DEFINITION;
```

プライベート一時表 (New in 18c)

- 一時的な、プライベート (session-based) 表定義
 - プライベートな表名と定義
- 一時的な、プライベート (session-based) データコンテンツ
 - セッション あるいは トランザクション中

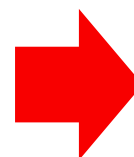


インライン外部表

外部データへの透過的アクセス

- 実行時に提供される外部表定義
- 一度のみ使われる外部表を事前に作成する必要なし
 - 開発者の生産性向上

```
-- 定義
CREATE TABLE sales_xt
  (prod_id number, ... )
  TYPE ORACLE_LOADER
  ...
  LOCATION 'new_sales_kw13')
  REJECT LIMIT UNLIMITED );
-- ロード
INSERT INTO sales SELECT * FROM sales_xt;
-- 削除
DROP TABLE sales_xt;
```



```
-- 定義 + ロード
INSERT INTO sales
SELECT sales_xt.*
FROM EXTERNAL (
  (prod_id number, ... )
  TYPE ORACLE_LOADER
  ...
  LOCATION 'new_sales_kw13')
  REJECT LIMIT UNLIMITED );
```


インライン外部表: WITH句を使った例

インライン・ビューと同様の感覚で、外部表をダイナミックに扱うことも可能

```
-- inline XT in WITH clause
with dept_xt as (
  select * from external(
    (deptno char(2), dname char(14), loc char(13))
    type oracle_loader
    default directory scott_def_dir1
    access parameters (fields terminated by ',')
    location ('deptexld01.dat')
    reject limit unlimited )
)
select d.dname from dept_xt d
where d.deptno = 10 order by 1;
```

列ベースの データ指向 Collation (照合) ANSI SQL:1999

- Oracle は、約100の照合をサポート: 言語ソート・比較
 - `_CI` あるいは `_AI` を足すことでパラメータ化
 - `_CI` - case-insensitive ソートを指定
 - `_AI` - Accent-insensitive ソートを指定

```
CREATE TABLE products
( product_code          VARCHAR2 (20 BYTE)
, product_name          VARCHAR2 (100 BYTE)
, product_category     VARCHAR2 (5 BYTE)
, product_description   VARCHAR2 (1000 BYTE)
);
```

```
COLLATE BINARY
COLLATE GENERIC_M_CI
COLLATE BINARY
COLLATE BINARY_CI
```

– *Product_name は GENERIC_M_CI を用いて比較される- case-insensitive 版の generic multilingual collation*

宣言された照合がある列のパーティション化

- 表のパーティション化キー列に、宣言された照合を持つことができるようになった
- 以前のOracle Databaseリリースでは、表のパーティション化キー列は、`BINARY`、`USING_NLS_COMP`、`USING_NLS_SORT`および`USING_NLS_SORT_CS`の照合順序のみに制限されていた
- 表のパーティション化キー列の大/小文字を区別しない照合を宣言することにより、大/小文字を区別しない方法で文字値で表をパーティション化できる

OCIバインド変数の照合

- 属性OCI_ATTR_COLLATION_IDの値を介して、問合せまたはDML文のバインド変数の照合を渡すことが可能になった
- バインド変数の照合を設定することにより、比較条件でのバインド変数の値と、バインド変数が引数であるその他のSQL操作を比較する照合を制御できる

Analyticビューの改良

- Analytic Viewsでの計算拡充:
 - ランキングと統計関数
 - RANK_*, PERCENTILE_*, STATS_*, COVAR_*
 - 階層表現
 - HIER_DEPTH, HIER_LEVEL, HIER_MEMBER_NAME, など
- 幅広いスキーマサポート:
 - スノーflakeスキーマ; フラット/非正規化ファクト表 (スタースキーマに加え)
- よりパワフルな SQL:
 - FILTER FACT: 集計前に分析ビューからアクセスされたデータをフィルタ
 - ADD MEASURES: 独自の計算済メジャーを動的に定義

テック・ナイトアーカイブ資料と お役立ち情報

各回テック・ナイトセッション資料 ダウンロードサイト

oracle technight



[技術コラム しば
ちよう先生の
試して納得！
DBAへの道](#)



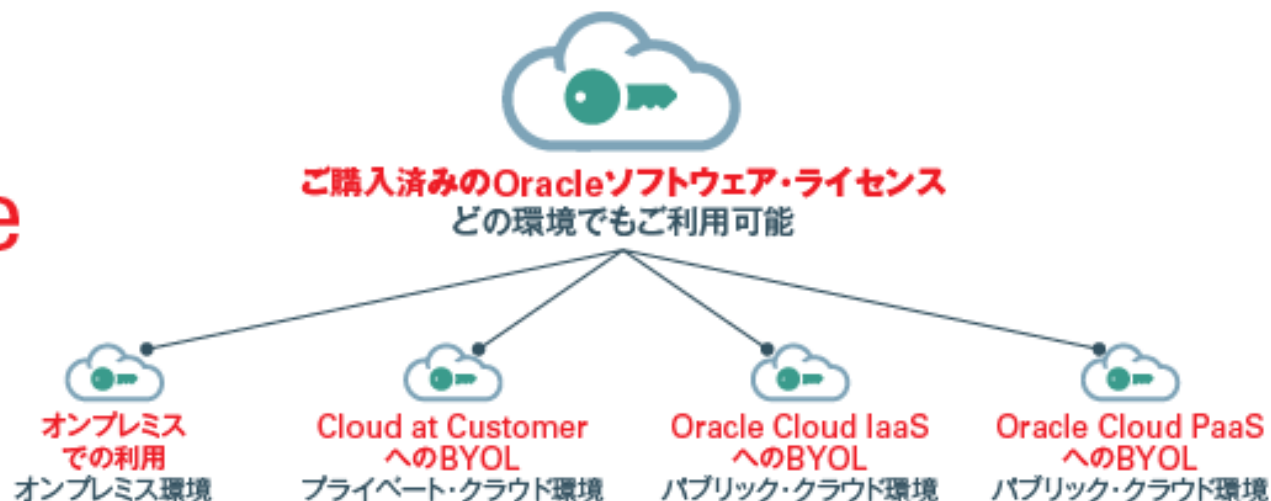
[技術コラム 津島
博士の
パフォーマンス
講座](#)



[もしも
みなみんなが
DBをクラウドで
動かしてみたら](#)

Bring Your Own License

既存のオラクル・ライセンスを柔軟にクラウド環境で活用



300ドル分の無料トライアルでOracle Cloudを体験!



https://cloud.oracle.com/ja_JP/tryit

Oracle Cloudでは各種クラウドサービスを300ドル分無料でお試しいただけるトライアルサービスをご提供しております。無料トライアルのお申込み方法の詳細は、左のQRコード、またはURLにアクセスしてください。

Oracle Cloudのユースケース、導入事例、資料、価格などの詳細情報は、下記URLにアクセスしてください。

<http://www.oracle.com/jp/cloud/platform/overview/index.html>

～ みなさまの投稿をお待ちしております ～



Twitter

#OracleTechNight

こんな時、かけこむ会社が増えています。



ビジネスプロセスを
改善したい!



今のシステムは
使いにくい!



システムコストを
下げたい!



パフォーマンスを
良くしたい!



経営分析を
したいのだが...



どんなソリューションが
あるの?



見積りはどれくらい
なんだろう?



楽に管理を
したい!

Oracle Digitalは、オラクル製品の導入をご検討いただく際の総合窓口。
電話とインターネットによるダイレクトなコミュニケーションで、どんなお問い合わせにもすばやく対応します。
もちろん、無償。どんなことでも、ご相談ください。



お問い合わせは電話またはWebフォーム

☎ 0120-155-096

受付時間 月～金 9:00-12:00 / 13:00-17:00
(祝日および年末年始休業日を除きます)

<http://www.oracle.com/jp/contact-us>

Integrated Cloud

Applications & Platform Services

ORACLE®