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

18c ORACLE Database

Oracle Database 18c テクノロジーシリーズ 2
「RAC/Sharding と Data Guard/HA の機能強化」
～ RAC/Sharding～

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
日下部 明

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。

アジェンダ

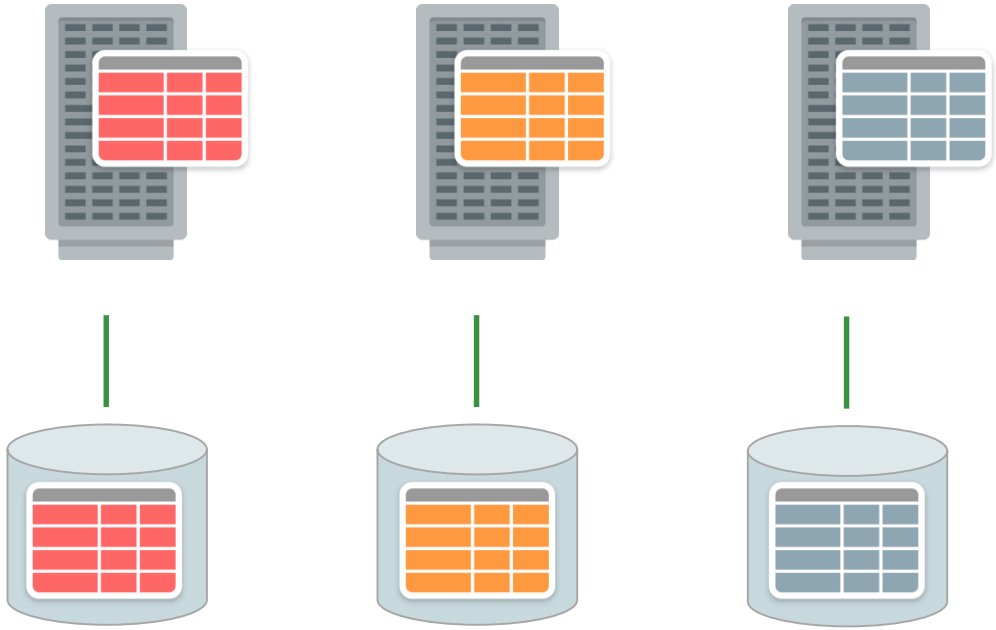
1. Sharding
2. Scalable SEQUENCE

Sharding

シャーディング

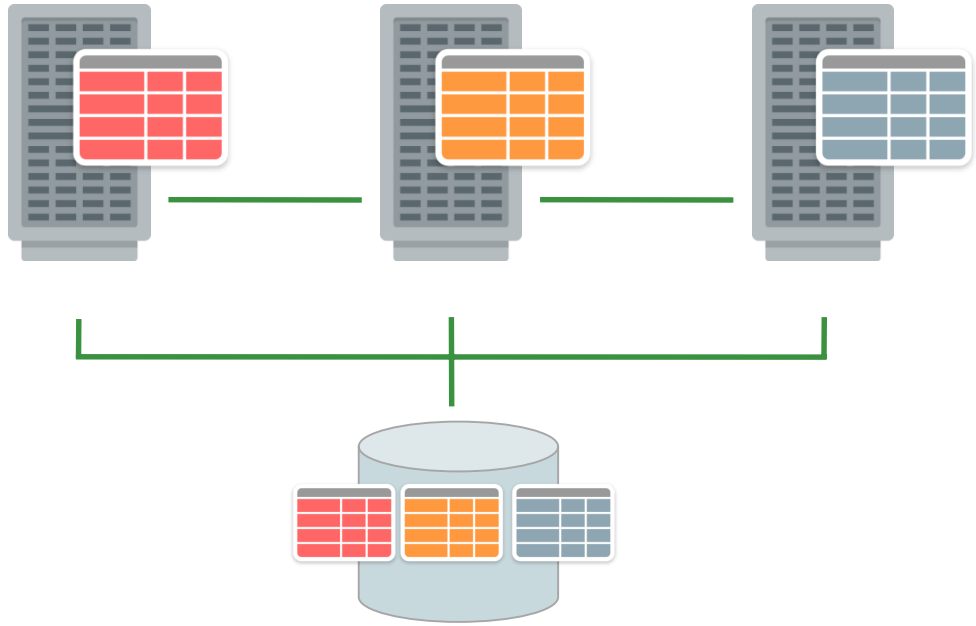
- データを水平分割する
- クライアント・アクセスを分散してスケーラビリティを得る
  - オンライン・トランザクション処理向け

2種類のシャーディング機能



Oracle Database Sharding（12.2～）

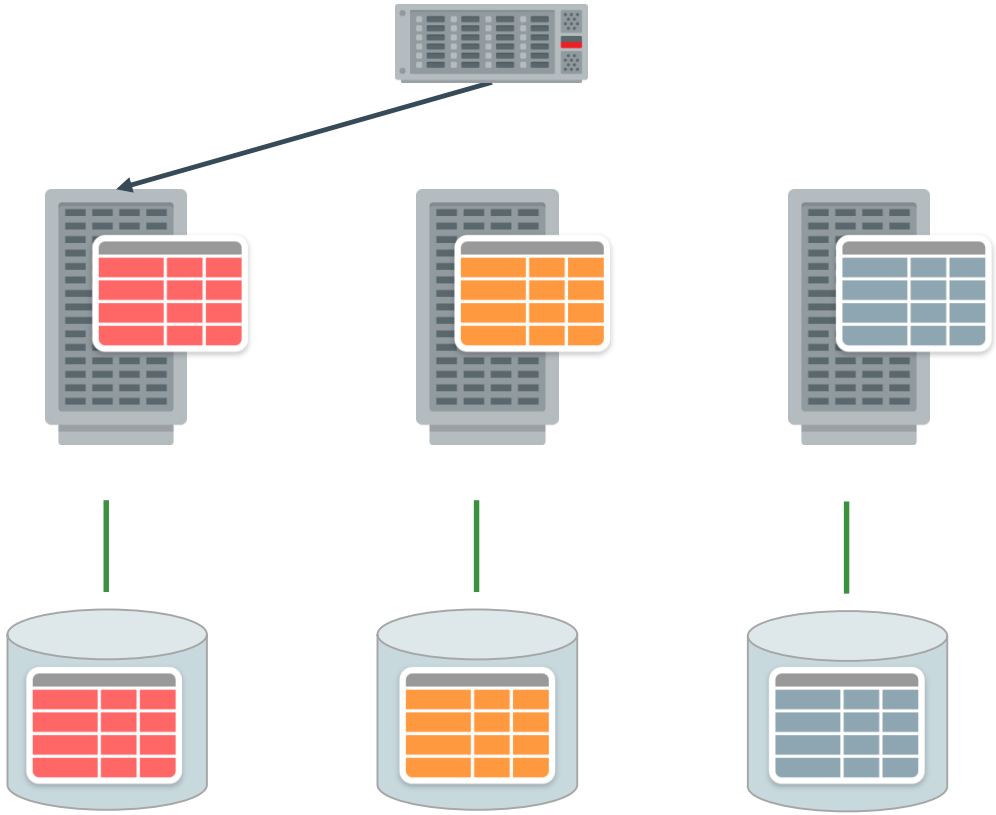
複数のデータベースに表パーティションを分散



Oracle RAC Sharding（18.1～）

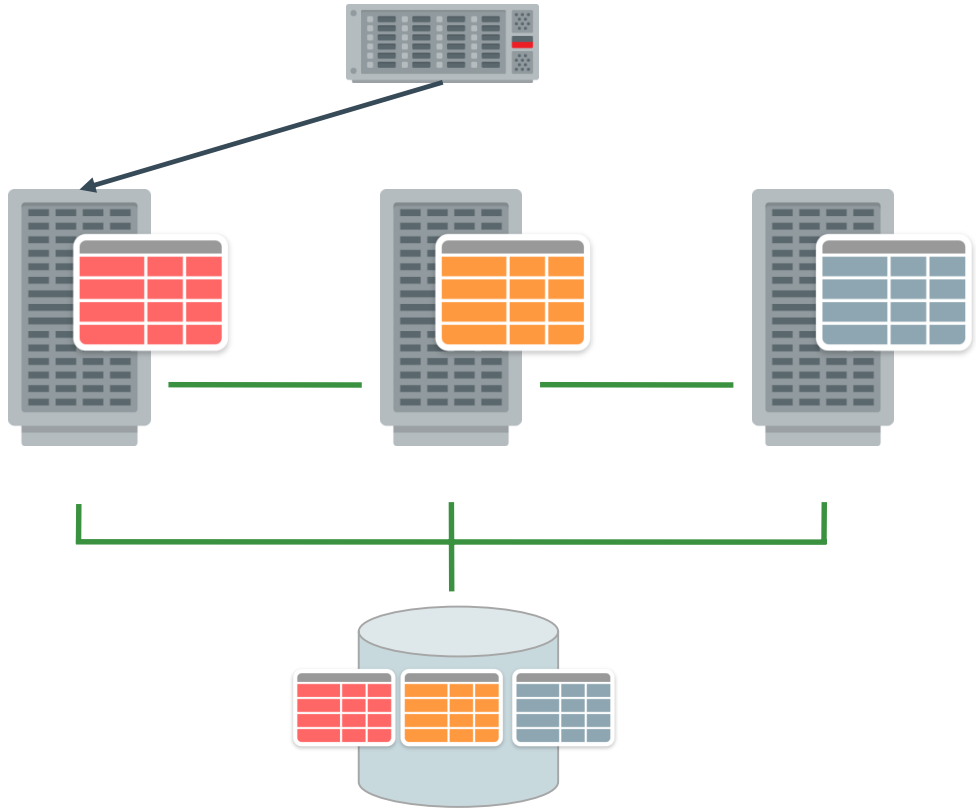
1つのデータベースの複数インスタンスに
表パーティションのアフィニティを持たせる

クライアントはシャーディング・キーを指定してアクセス



Oracle Database Sharding（12.2～）

複数のデータベースに表パーティションを分散



Oracle RAC Sharding（18.1～）

1つのデータベースの複数インスタンスに
表パーティションのアフィニティを持たせる

Javaのコード：従来のコネクション取得方法

- データ・ソース（UCP）のgetConnection()メソッドを呼ぶ

```
Connection con = ods.getConnection();       // コネクションをプールから取得

    ... // SQL実行

con.commit();
con.close();                                // コネクションをプールに返却
```

Javaのコード：シャーディング・キーを指定

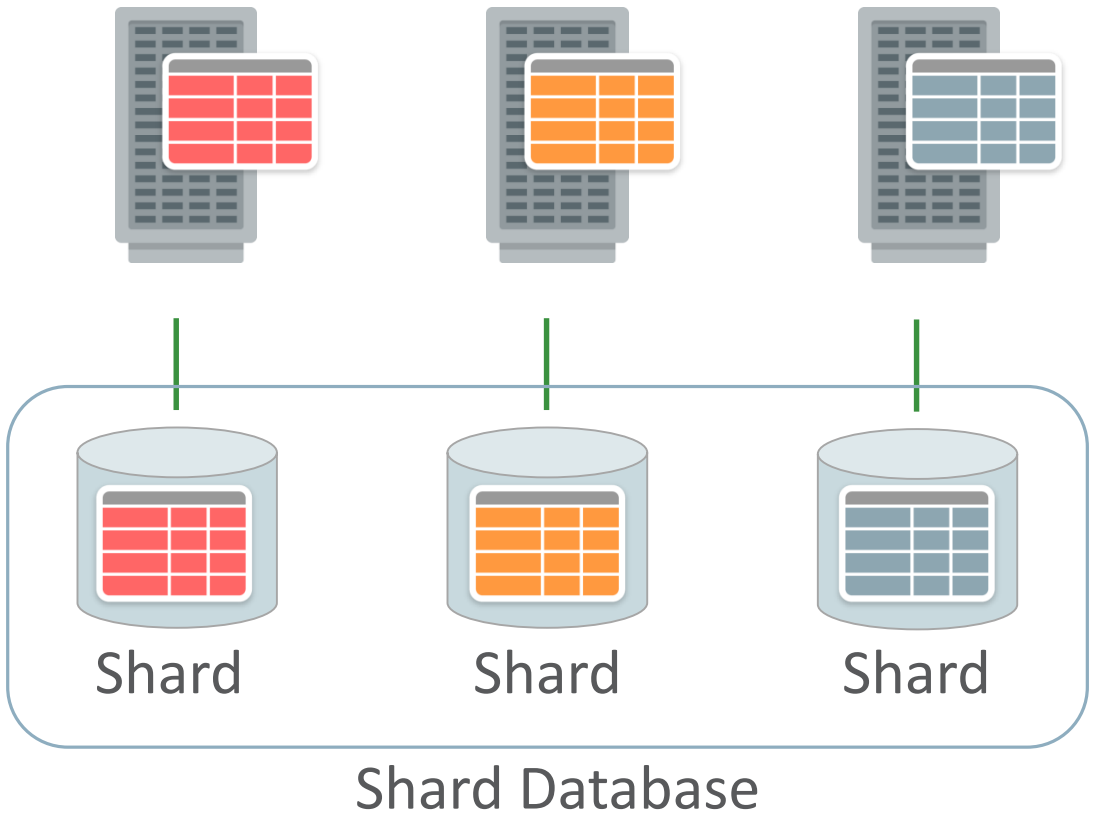
- シャーディング・キーを指定してbuild()メソッドを呼ぶ

```
Date shardingKeyVal = new java.sql.Date(0L);  // シャーディング・キーになる値
OracleShardingKey sdkey = ods.createShardingKeyBuilder()
                                .subkey(shardingKeyVal, OracleType.DATE)
                                .build();          // シャーディング・キーを生成

Connection con = ods.createConnectionBuilder()
                        .shardingKey(sdkey)
                        .build();  // シャーディング・キーを指定してコネクションをプールから取得

    ... // SQL実行

con.commit();
con.close();                                       // コネクションをプールに返却
```

Oracle Database Sharding

必要なエディション：
DBCS-EE&HP（Shard 3つまで）
EE, EE-ES, DBCS-EP, ExaCS（Shard数制限無し）

Oracle Database Sharding



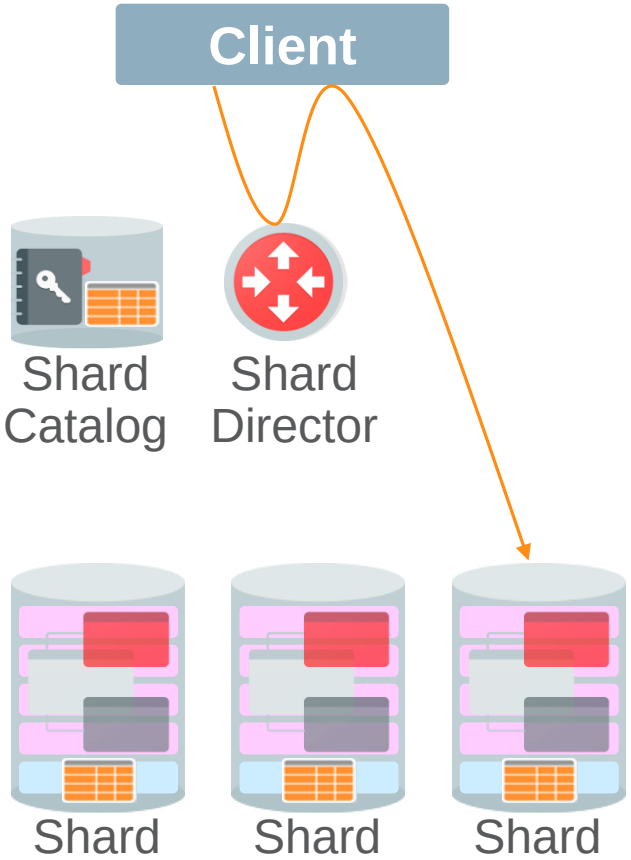
- 複数のDBサーバーで論理DBを構成
  - 水平分割
  - パーティション・キーを指定
- DBサーバー間で共有ハードウェアを持たない
- 1つのデータベースをシャードと呼ぶ
- 全体はシャード・データベースと呼ぶ

Shardingの構成要素

業務ロジック・
DBへの接続リクエスト

Shardingの構成・管理
フレームワーク

アプリケーションの
データを分散配置する
DB群(Shard)

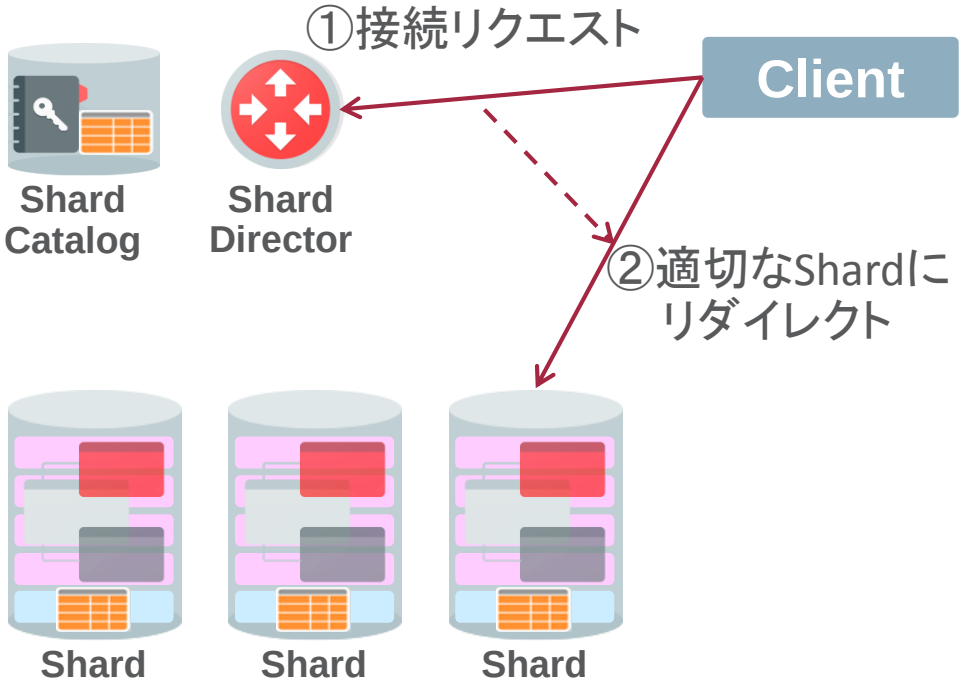


Shard Catalog
Shardingの構成情報や構成・管理タスクを保持するリポジトリDB

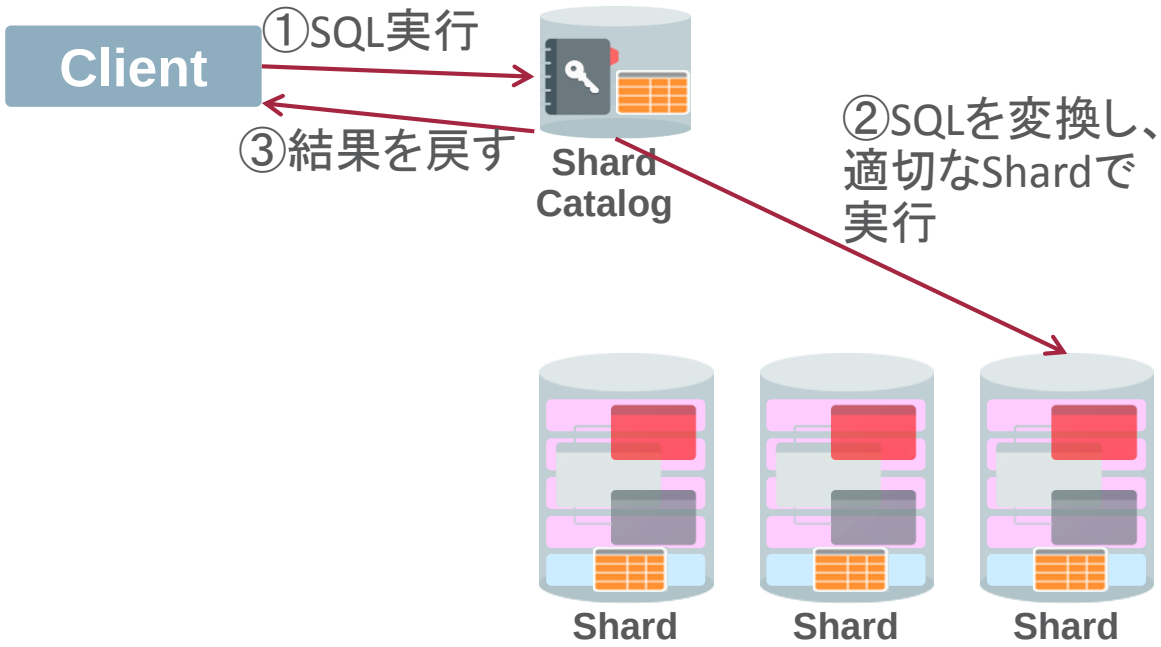
Shard Director
各Shardに対する構成・管理処理の実行
アプリケーション接続ルーティングを行う
インスタンス(OSプロセス群+共有メモリ)

アプリケーションから Oracle Database Sharding への接続

- Direct Routing
  - Sharding Key を指定してShard Directorからのリダイレクトで適切なShardに直接接続する



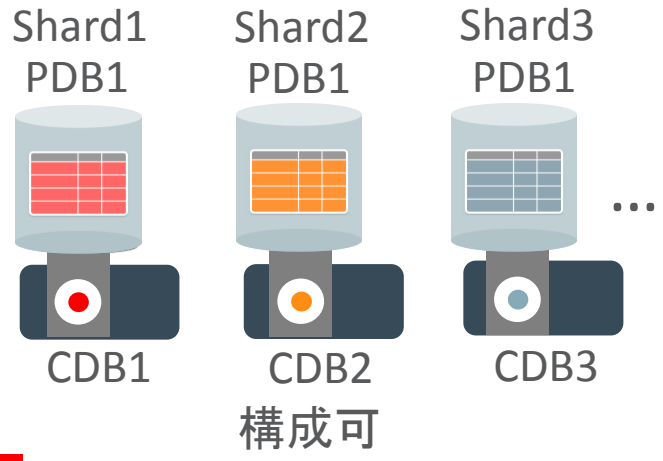
- Proxy Routing
  - ClientからShard CatalogにSQL実行
  - Shard Catalogは、SQLから適切なShardを判断し実行、結果をClientに戻す

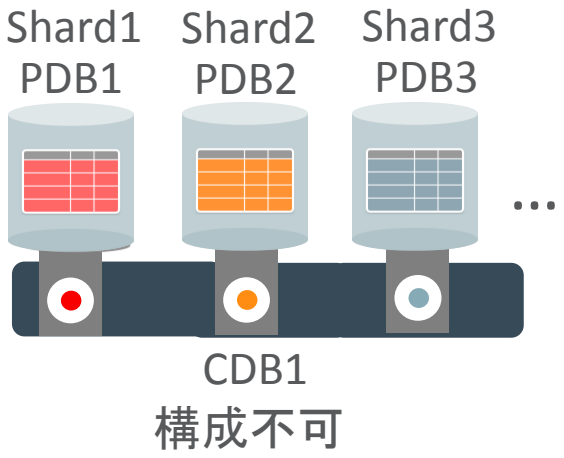


シャードに使用できるデータベースの構成

Multitenant Support

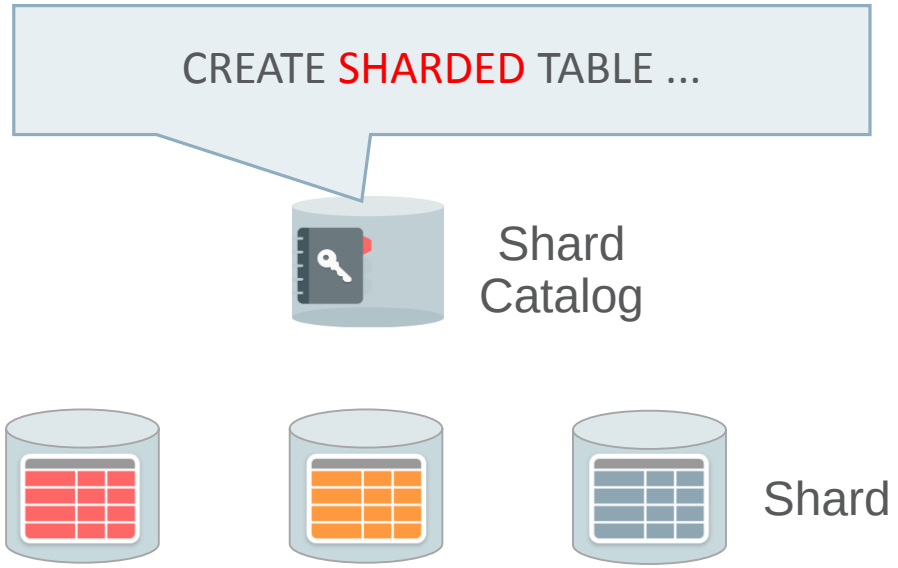
- 12c まではシャードとして使えるのは non-CDB のみ
- 18c からはシャードとして PDB が使える
  - ただし、Oracle Database Sharding構成に組み込むことができるPDBは1つのCDBにつき1つ
  - CDBは他の非Shard PDBを含むことができる



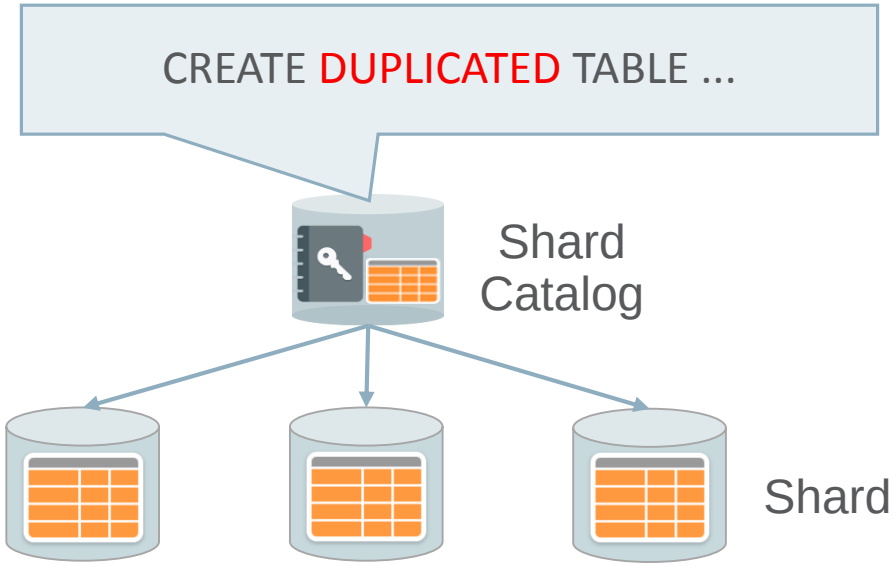


データの配置

- シャード表
  - パーティション表
  - パーティション・キーで水平分割
  - 各シャードで排他な内容を持つ



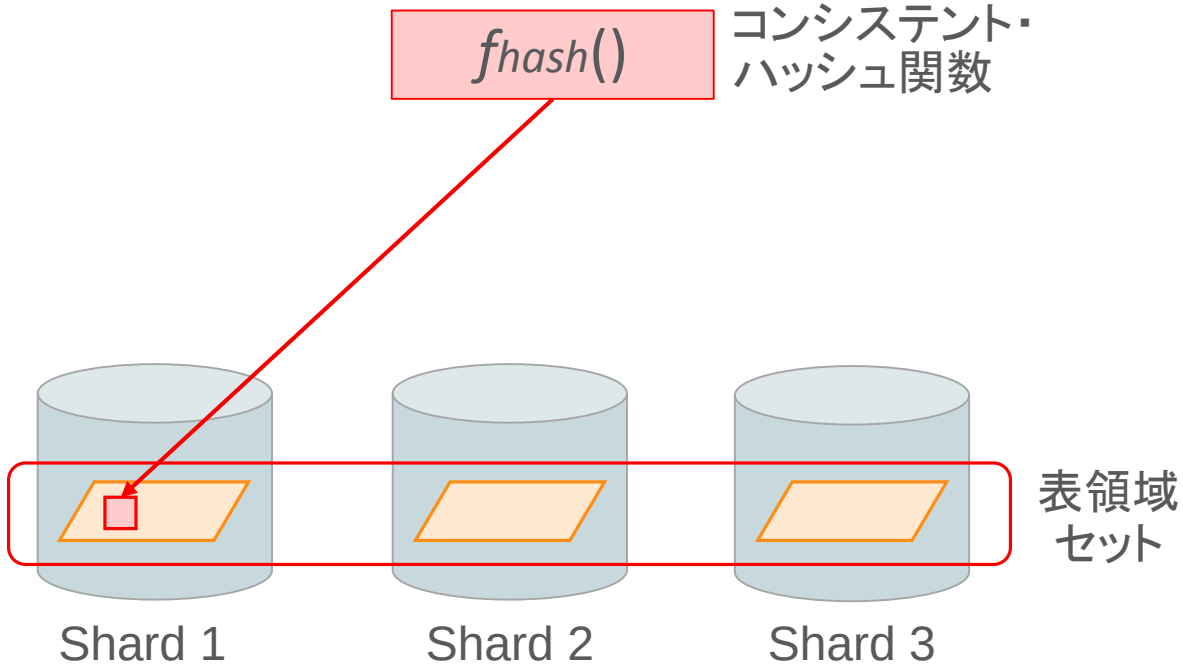
- 重複表
  - パーティション化できない表
  - すべてのシャードで同じ内容を持つ
    - シャード・カタログにも同じ内容を持つ
    - マテリアライズド・ビューで伝搬



シャード表のパーティショニング方式

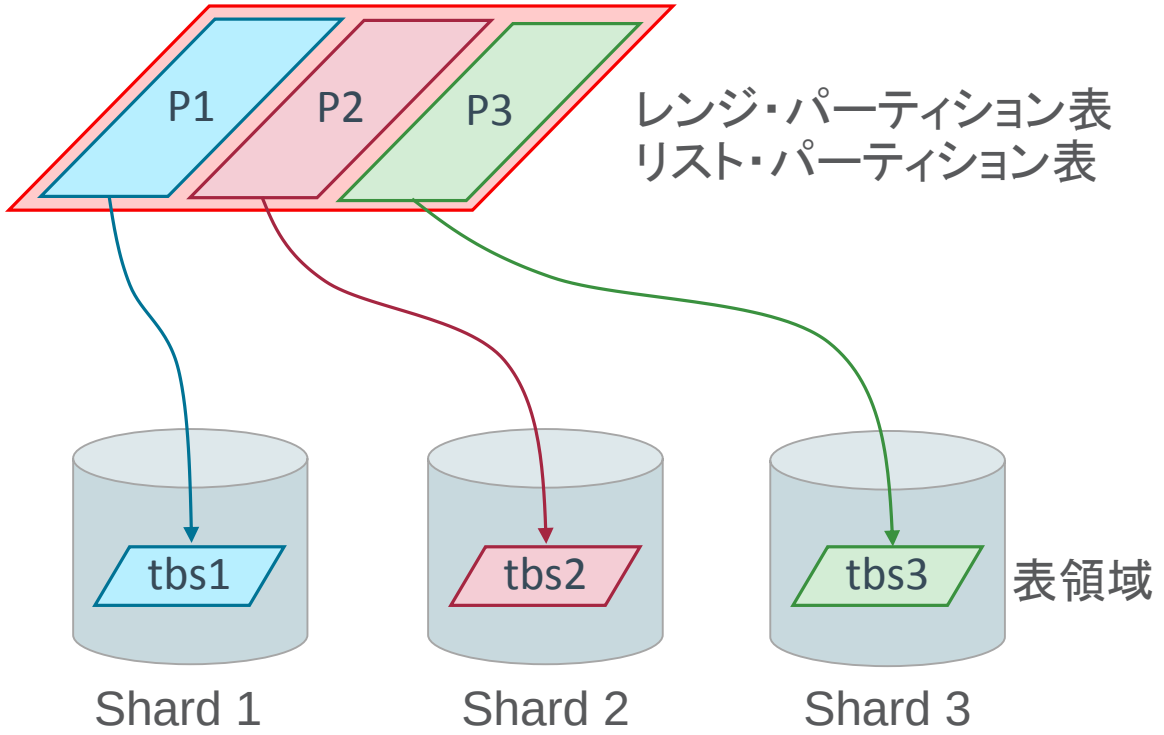
- 12.2
  - システム管理シャーディング
    - 各シャードへのデータの配置はシステムが決める
      - コンシステント・ハッシュ
  - コンポジット・シャーディング
    - レンジ-コンシステント・ハッシュ
    - リスト-コンシステント・ハッシュ
- 18～
  - ユーザー定義シャーディング
    - 特定のパーティションを特定のシャードに対応付ける
      - レンジ（ただしインターバルは不可）
      - リスト

シャード表の配置



システム管理シャーディング

1. 表領域セットを構成
2. システムがパーティション配置を決定



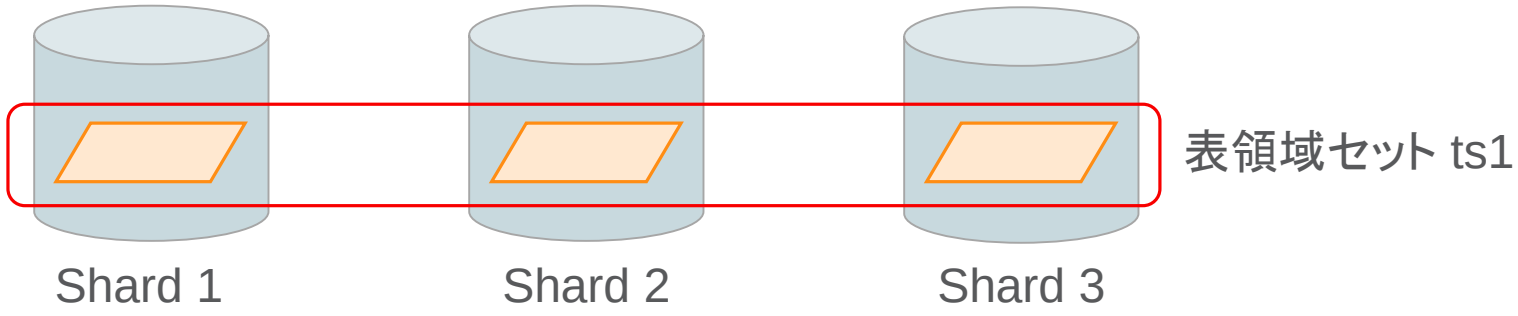
ユーザー定義シャーディング

1. 特定の表領域を特定のシャードに構成
2. 特定のパーティションを特定の表領域に配置

システム管理シャーディング

複数のシャードにまたがる表領域セットを構成

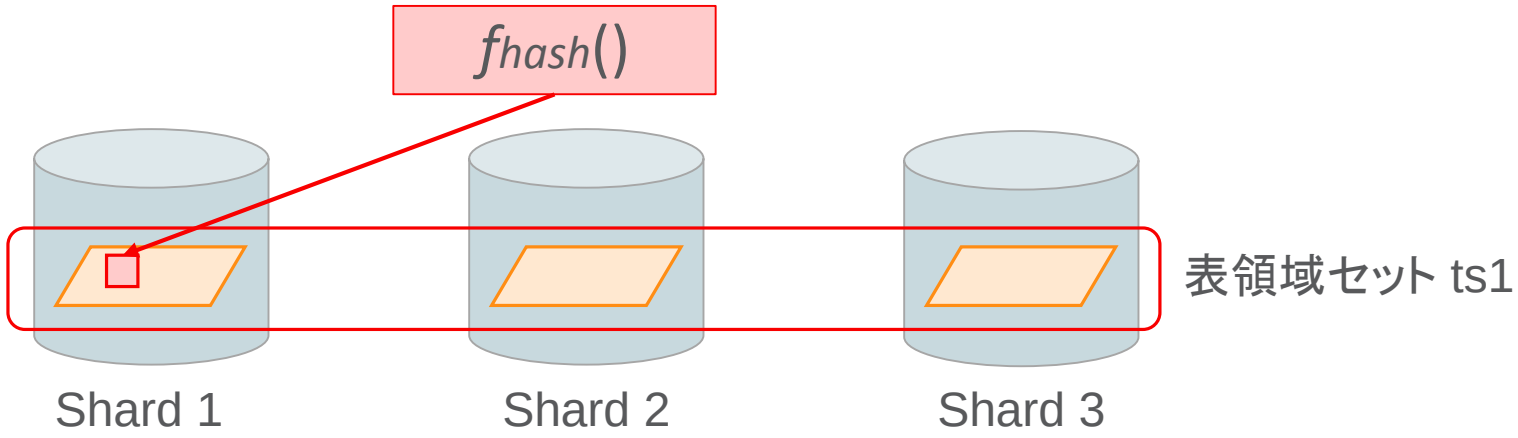
```
CREATE TABLESPACE SET ts1
USING TEMPLATE
(
 DATAFILE SIZE 10M
 EXTENT MANAGEMENT LOCAL UNIFORM SIZE 256K
 SEGMENT SPACE MANAGEMENT AUTO
 ONLINE
);
```



システム管理シャーディング

コンシステント・ハッシュ表

```
CREATE SHARDED TABLE customers
( cust_id     NUMBER NOT NULL
:
, CONSTRAINT cust_pk PRIMARY KEY(cust_id)
)
PARTITION BY CONSISTENT HASH (cust_id)
PARTITIONS AUTO
TABLESPACE SET ts1
;
```



ユーザー定義シャーディングのメリット - 地理分散

リニアなスケーラビリティ / 計画停止の制御

ユーザー定義シャーディング

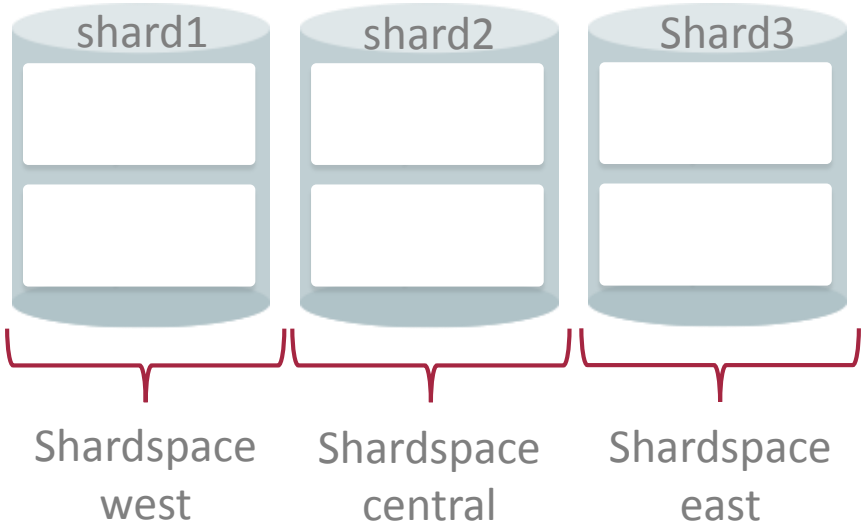
シャード領域と表領域を対応付ける

- GDSCTLコマンドを使ってシャード領域を作成

```
GDSCTL> ADD SHARDSPACE -SHARDSPACE west;
GDSCTL> ADD SHARDSPACE -SHARDSPACE central;
GDSCTL> ADD SHARDSPACE -SHARDSPACE east;
GDSCTL> ADD SHARD -CONNECT shard1 -SHARDSPACE west;
GDSCTL> ADD SHARD -CONNECT shard2 -SHARDSPACE central;
GDSCTL> ADD SHARD -CONNECT shard3 -SHARDSPACE east;
```

- シャード領域を指定して表領域を作成

```
CREATE TABLESPACE tbs1 IN SHARDSPACE west;
CREATE TABLESPACE tbs2 IN SHARDSPACE west;

CREATE TABLESPACE tbs3 IN SHARDSPACE central;
CREATE TABLESPACE tbs4 IN SHARDSPACE central;

CREATE TABLESPACE tbs5 IN SHARDSPACE east;
CREATE TABLESPACE tbs6 IN SHARDSPACE east;
```



ユーザー定義シャーディング

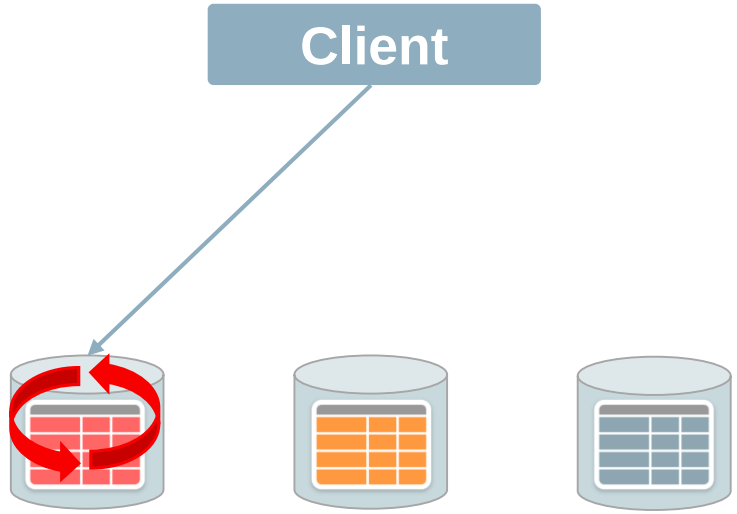
レンジ/リスト・パーティション表

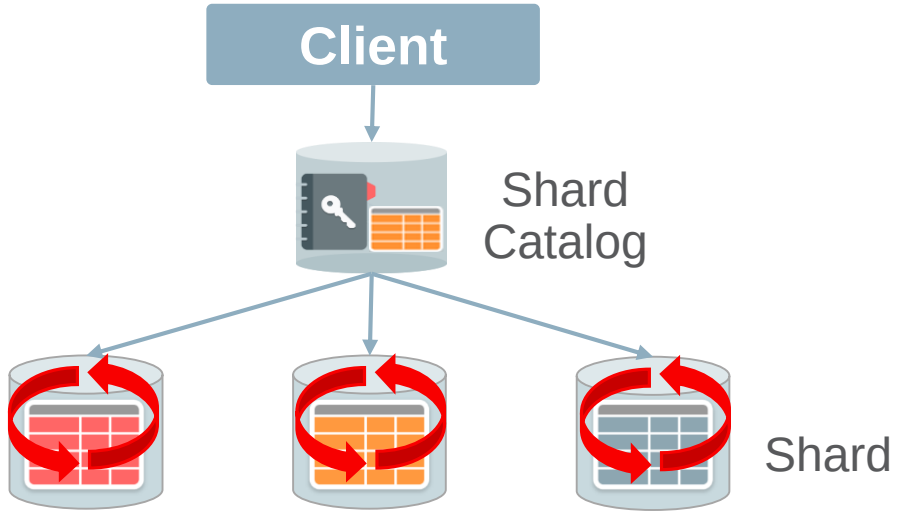
- 各パーティションを格納する表領域を指定

```
CREATE SHARDED TABLE accounts
( id              NUMBER,
  account_number NUMBER,
  customer_id     NUMBER,
  branch_id       NUMBER,
  state           VARCHAR(2) NOT NULL,
  status          VARCHAR2(1)
)
PARTITION BY LIST (state)
( PARTITION p_northwest VALUES ('OR', 'WA') TABLESPACE tbs1,
  PARTITION p_southwest VALUES ('AZ', 'UT', 'NM') TABLESPACE tbs2,
  PARTITION p_northcentral VALUES ('SD', 'WI') TABLESPACE tbs3,
  PARTITION p_southcentral VALUES ('OK', 'TX') TABLESPACE tbs4,
  PARTITION p_northeast VALUES ('NY', 'VM', 'NJ') TABLESPACE tbs5,
  PARTITION p_southeast VALUES ('FL', 'GA') TABLESPACE tbs6
)
```

シャード表に対するSQL実行

- あるシャードから別のシャードは参照できない
  - スケーラビリティを得るためにはDirect Routingを想定
- シャード・カタログからは全シャードのデータを参照できる
  - マルチシャード・クエリ





シャード表の参照

- 特定のシャードで実行
  - そのシャードにある行のみ見える

```
SQL> select * from accounts;

        ID ACCOUNT_NUMBER CUSTOMER_ID  BRANCH_ID ST S
---------- -------------- ----------- ---------- -- -
         1              1           1          1 OR Y
         2              2           2          2 WA Y
         3              3           3          3 AZ Y
         4              4           4          4 UT Y
         5              5           5          5 NM Y
```

- シャード・カタログで実行
  - すべての行が見える

```
SQL> select * from accounts;

        ID ACCOUNT_NUMBER CUSTOMER_ID  BRANCH_ID ST S
---------- -------------- ----------- ---------- -- -
         1              1           1          1 OR Y
         2              2           2          2 WA Y
         3              3           3          3 AZ Y
         4              4           4          4 UT Y
         5              5           5          5 NM Y
         6              6           6          6 SD Y
         7              7           7          7 WI Y
         8              8           8          8 OK Y
         9              9           9          9 TX Y
        10             10          10         10 NY Y
        11             11          11         11 VM Y
        12             12          12         12 NJ Y
        13             13          13         13 FL Y
        14             14          14         14 GA Y
```

シャード表の参照

シャード・カタログからのデータの参照

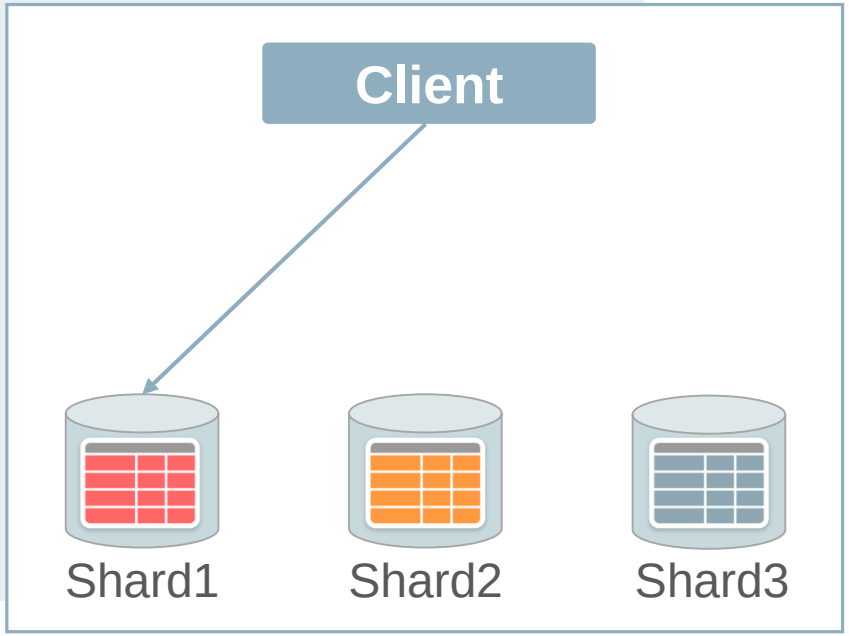
- 停止しているシャードがある場合にはマルチシャード・クエリは失敗する

```
SQL> select * from accounts;
select * from accounts
*
ERROR at line 1:
ORA-02519: cannot perform cross-shard operation. Chunk "3" is unavailable
ORA-06512: at "GSMADMIN_INTERNAL.DBMS_GSM_POOLADMIN", line 21843
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 86
ORA-06512: at "GSMADMIN_INTERNAL.DBMS_GSM_POOLADMIN", line 21808
ORA-06512: at "GSMADMIN_INTERNAL.DBMS_GSM_POOLADMIN", line 21860
ORA-06512: at line 1
```

シャード表の更新

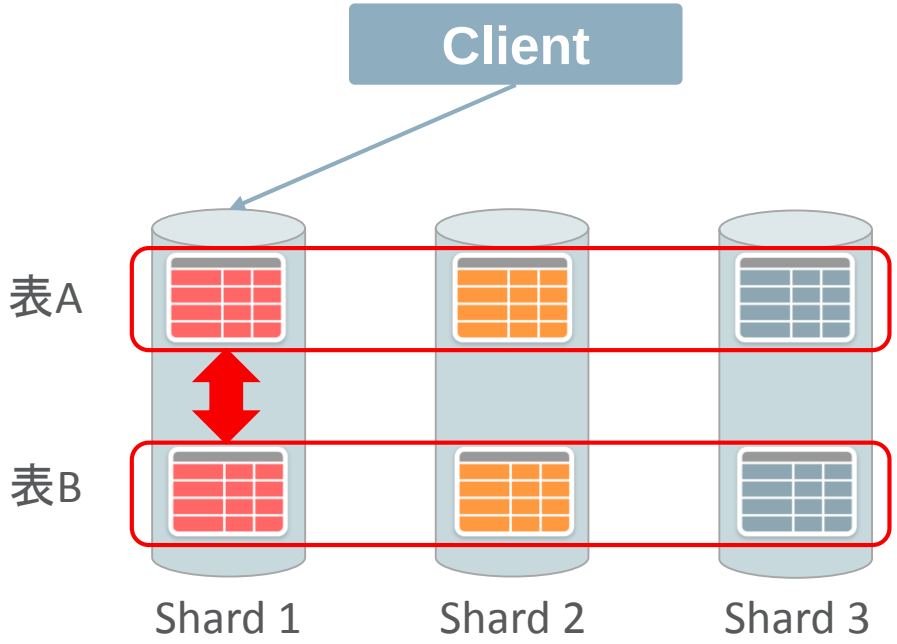
- 別のシャードに対するデータは格納出来ない（下記はShard1で実行）

```
### Shard1 に配置されるはずのデータをShard1でINSERT
SQL> insert into accounts values (1,1,1,1,'OR','Y');

1行が作成されました。

SQL> commit;

コミットが完了しました。

### Shard3 に配置されるはずのデータをShard1でINSERT
SQL> insert into accounts values (14,14,14,14,'GA','Y');
insert into accounts values (14,14,14,14,'GA','Y')
            *
行1でエラーが発生しました。:
ORA-14466:
読取り専用のパーティションまたはサブパーティション内のデータは変更できません。
```



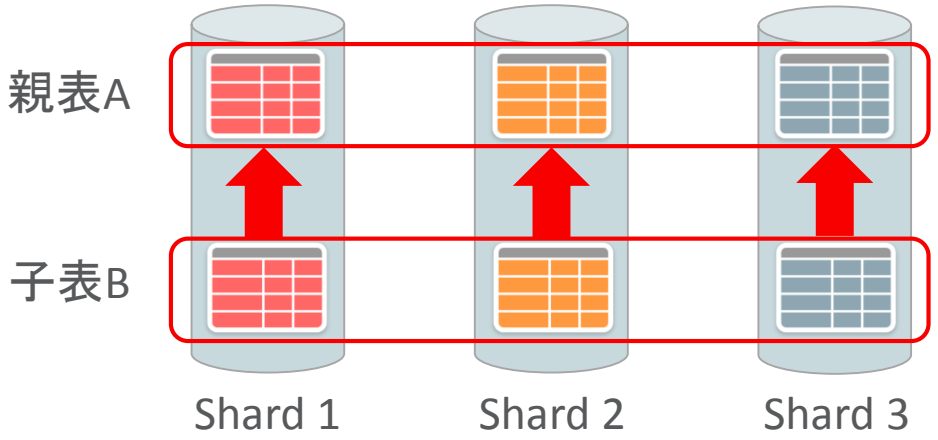
シャード表のジョイン

- KVSモデルとは異なりRDMBSでは表のジョインという概念がある
- ジョインする対象が同じシャードに配置されている必要がある



表ファミリ

- 複数のシャード表の間に親子関係を持たせる
  - システム管理シャーディングの親表のパーティションと子表のパーティションを同じシャードに配置する
  - 親子関係がある一連のシャード表を表ファミリと呼ぶ
- 親を持たない表をルート表と呼ぶ
  - 複数のルート表を持つことはできない
- 親子関係の定義
  - リファレンス・パーティション
  - PARENT句



表ファミリ

リファレンス・パーティション - 外部キーで親表と同じパーティションの区切り方

親表

- プライマリ・キーとパーティション・キーがCustNo列

```
CREATE SHARDED TABLE Customers
( CustNo       NUMBER NOT NULL
, Name         VARCHAR2(50)
, Address      VARCHAR2(250)
, CONSTRAINT RootPK PRIMARY KEY(CustNo)
)
PARTITION BY CONSISTENT HASH (CustNo)
PARTITIONS AUTO
TABLESPACE SET ts1
;
```

子表

- 外部キーが親表のパーティション・キー
- 外部キーでリファレンス・パーティション

```
CREATE SHARDED TABLE Orders
( OrderNo    NUMBER NOT NULL
, CustNo     NUMBER NOT NULL
, OrderDate  DATE
, CONSTRAINT OrderPK PRIMARY KEY (CustNo,
  OrderNo)
, CONSTRAINT CustFK  FOREIGN KEY (CustNo)
  REFERENCES Customers(CustNo)
)
PARTITION BY REFERENCE (CustFK)
;
```

表ファミリ

PARENT句 - 明示的に親子関係を定義

親表

- パーティション・キーがCustNo列

```
 CREATE SHARDED TABLE Customers
( CustNo       NUMBER NOT NULL
, Name         VARCHAR2(50)
, Address      VARCHAR2(250)
, region       VARCHAR2(20)
, class        VARCHAR2(3)
, signup       DATE
)
PARTITION BY CONSISTENT HASH (CustNo)
PARTITIONS AUTO
TABLESPACE SET ts1
;
```

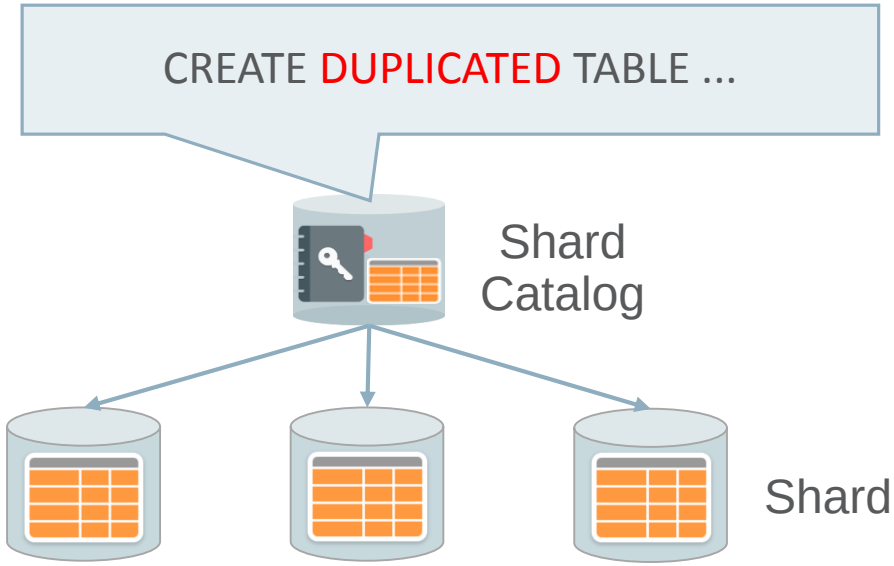
子表

- 親表と同じパーティション・キー
- PARENT句で親表の名前を指定

```
CREATE SHARDED TABLE Orders
( OrderNo    NUMBER
, CustNo     NUMBER NOT NULL
, OrderDate  DATE
)
PARENT Customers
PARTITION BY CONSISTENT HASH (CustNo)
PARTITIONS AUTO
TABLESPACE SET ts1
;
```

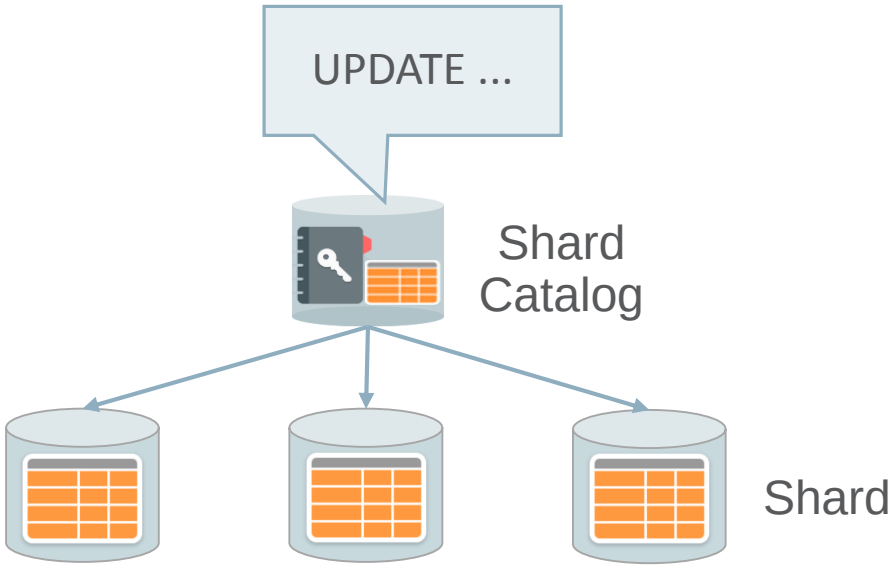
重複表

- 必ずしもすべての表がシャード表にできるとは限らない
  - シャード表はシャードをまたがったジョインができない
- すべてのシャードで同じ内容を持つ表を構成できる
  - シャード・カタログにある内容をマテリアライズド・ビューでシャードに伝搬

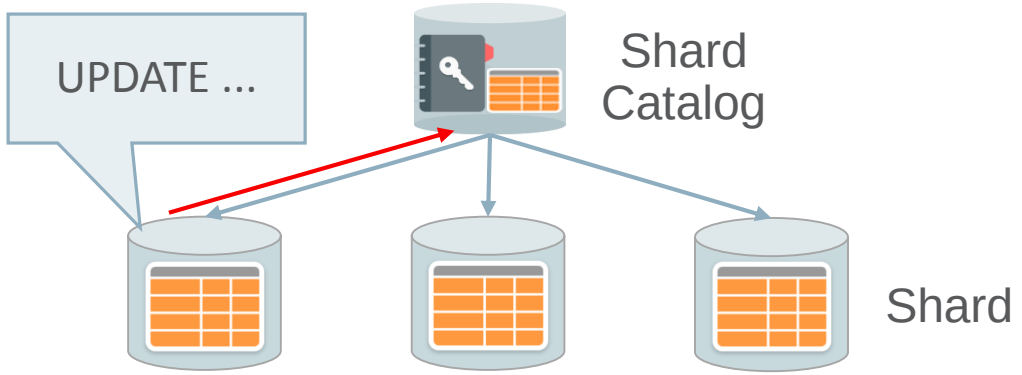


重複表の更新

- 12.2
  - シャード・カタログ上でのみ更新可能
  - 各シャードに伝搬



- 18～
  - 各シャード上でも更新可能
  - シャード・カタログで更新され各シャードに伝搬

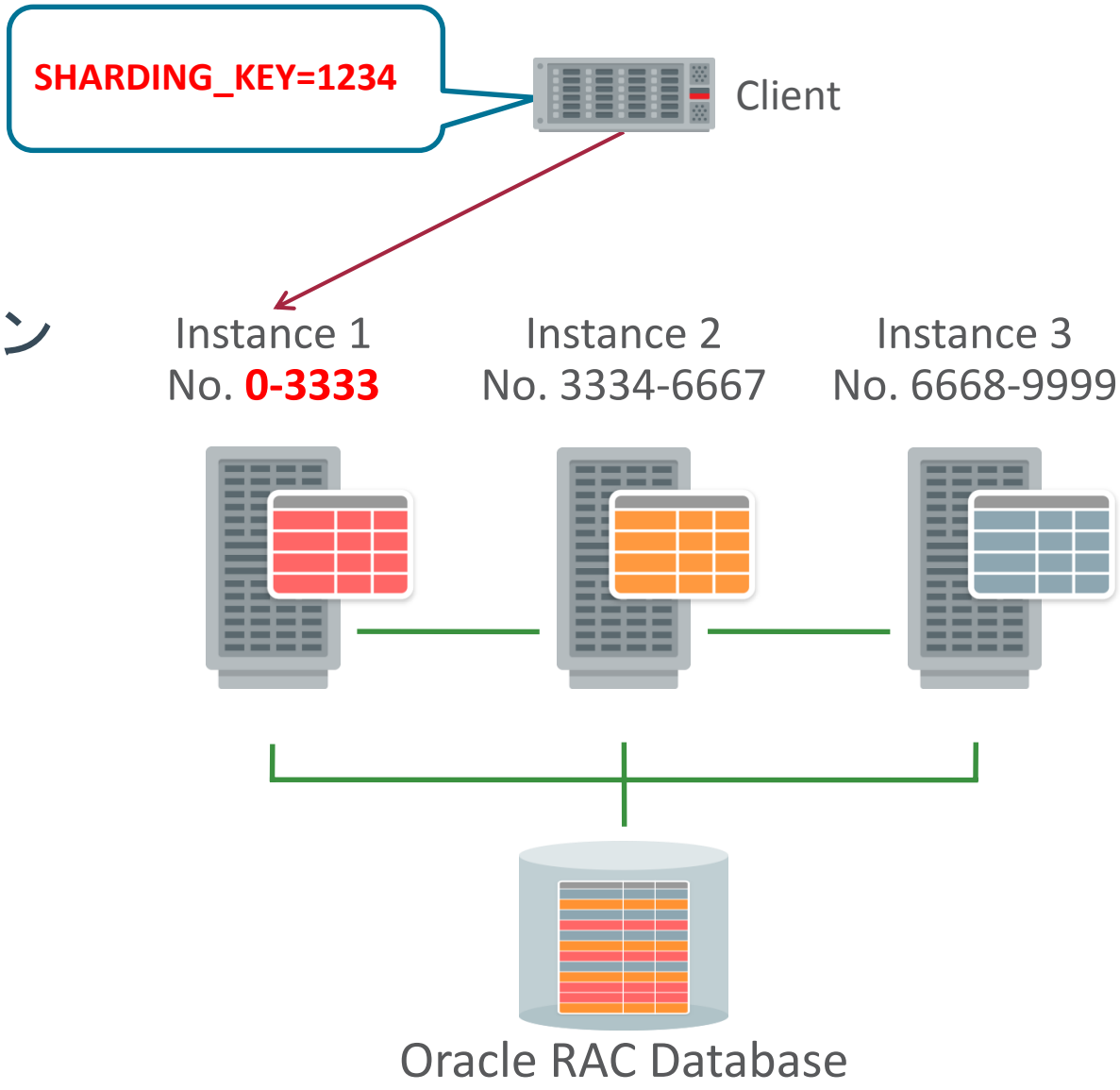


Oracle RAC Sharding

必要なエディション：RACと同様

Oracle RAC Sharding

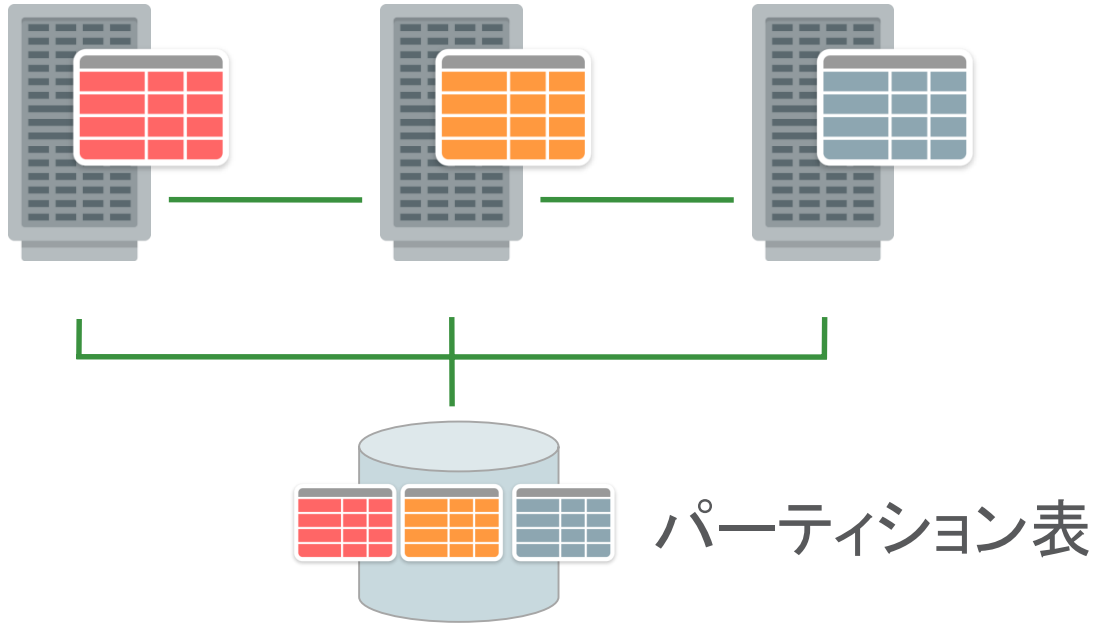
- アプリケーション側で定義されたキー（Sharding Key）によって接続するインスタンスを自動的に決定
- 通常のRACデータベース構成における論理的なパーティショニング
  - データは全インスタンスからアクセス可能
  - 各インスタンスのキャッシュの効率利用
  - Cache Fusion、インスタンス間競合の減少



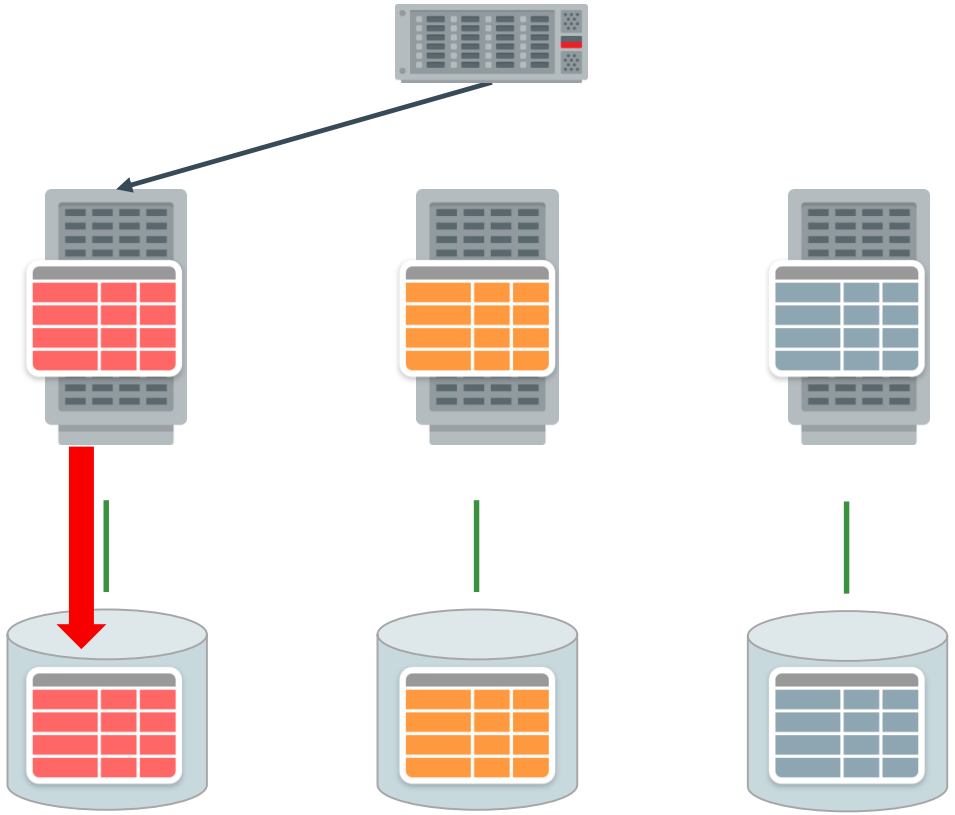
Oracle RAC Sharding

- パーティション表のパーティションにキャッシュのアフィニティを持たせる

```
SQL> ALTER SYSTEM ENABLE AFFINITY [ schema.]table [SERVICE service_name ];
```

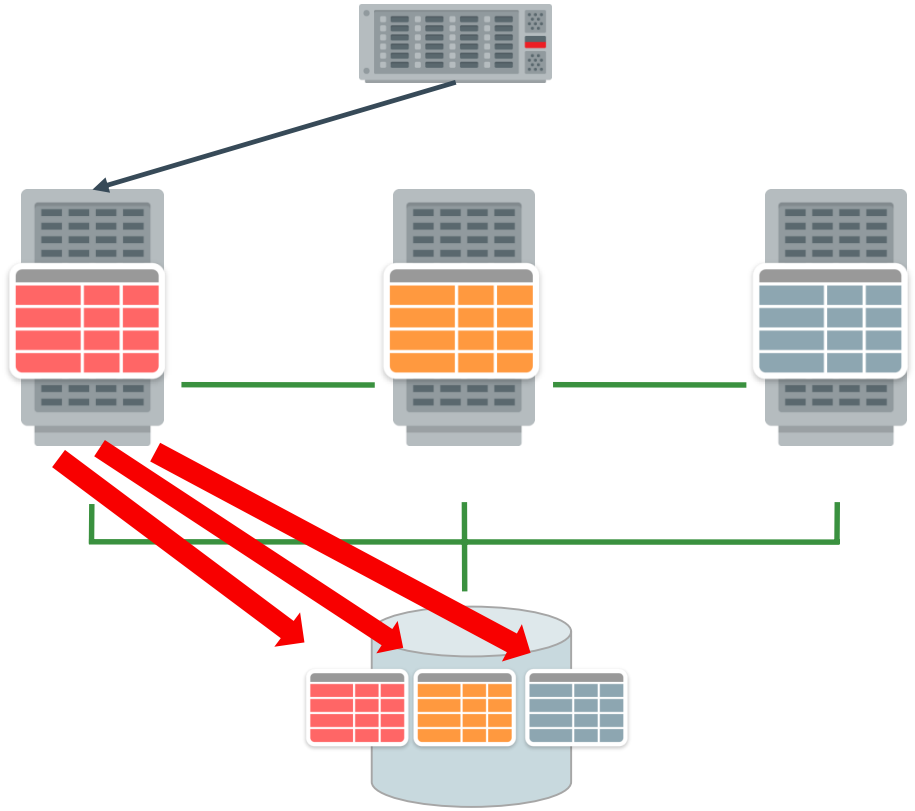


データのアクセス範囲



Oracle Database Sharding（12.2～）

特定のシャードのみ
（パーティションをまたがるジョインは不可）



Oracle RAC Sharding（18.1～）

複数のパーティションにまたがるアクセスは可能
（シングル・インスタンスと変わらない）

Sharding

- データの水平分割でクライアント・アクセスを分散
- オンライン・トランザクション処理向け
- シャーディング・キーを指定したアクセス

	Oracle Database Sharding	Oracle RAC Sharding
データベース構成	複数のデータベースの集合	1つのデータベース
スキーマ構成	シャード表/重複表	パーティション表/非パーティション表
データのアクセス範囲	特定のシャードのみ	すべてのデータにアクセス可能
停止の影響	シャードの停止が別のシャードに影響しない	インスタンス障害時わずかな時間ほかのインスタンスに影響する

リファレンス

- Oracle Database 12c Release 2
  - 「データベース管理者ガイド」
    - 第VII部 シャード・データベースの管理
- Oracle Database 18c
  - シャーディングに関するマニュアルが独立
  - 「Oracle Shardingの使用」

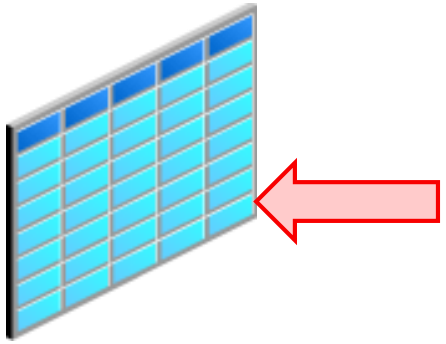
Scalable SEQUENCE

SEQUENCEのよくある使い方

- PRIMARY KEYなどに使用する一意な値を採番する

```
SQL> CREATE TABLE orders(
       orderid  NUMBER,
                :
       CONSTRAINT orders_pk PRIMARY KEY (orderid)
    );
```

orders表



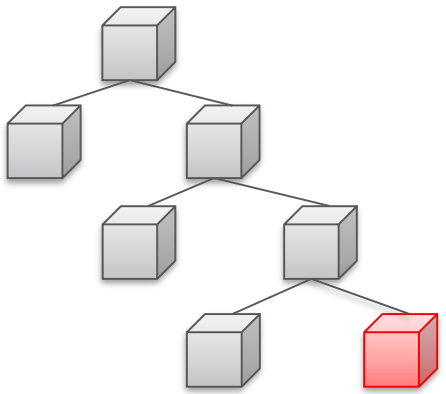
INSERT INTO orders VALUES (seq.NEXTVAL, ...)

索引の付いている列にSEQUENCEで生成した値をINSERT

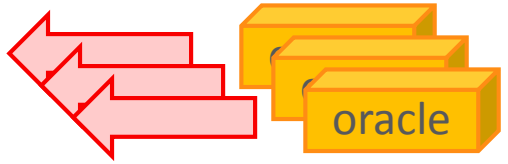
- B*ツリー索引は列の値を昇順に格納している
  - 最も大きな値を格納するリーフ・ブロックに更新が集中する

```
SQL> CREATE TABLE orders(
       orderid  NUMBER,
                :
       CONSTRAINT orders_pk PRIMARY KEY (orderid)
    );
```

B*ツリー索引



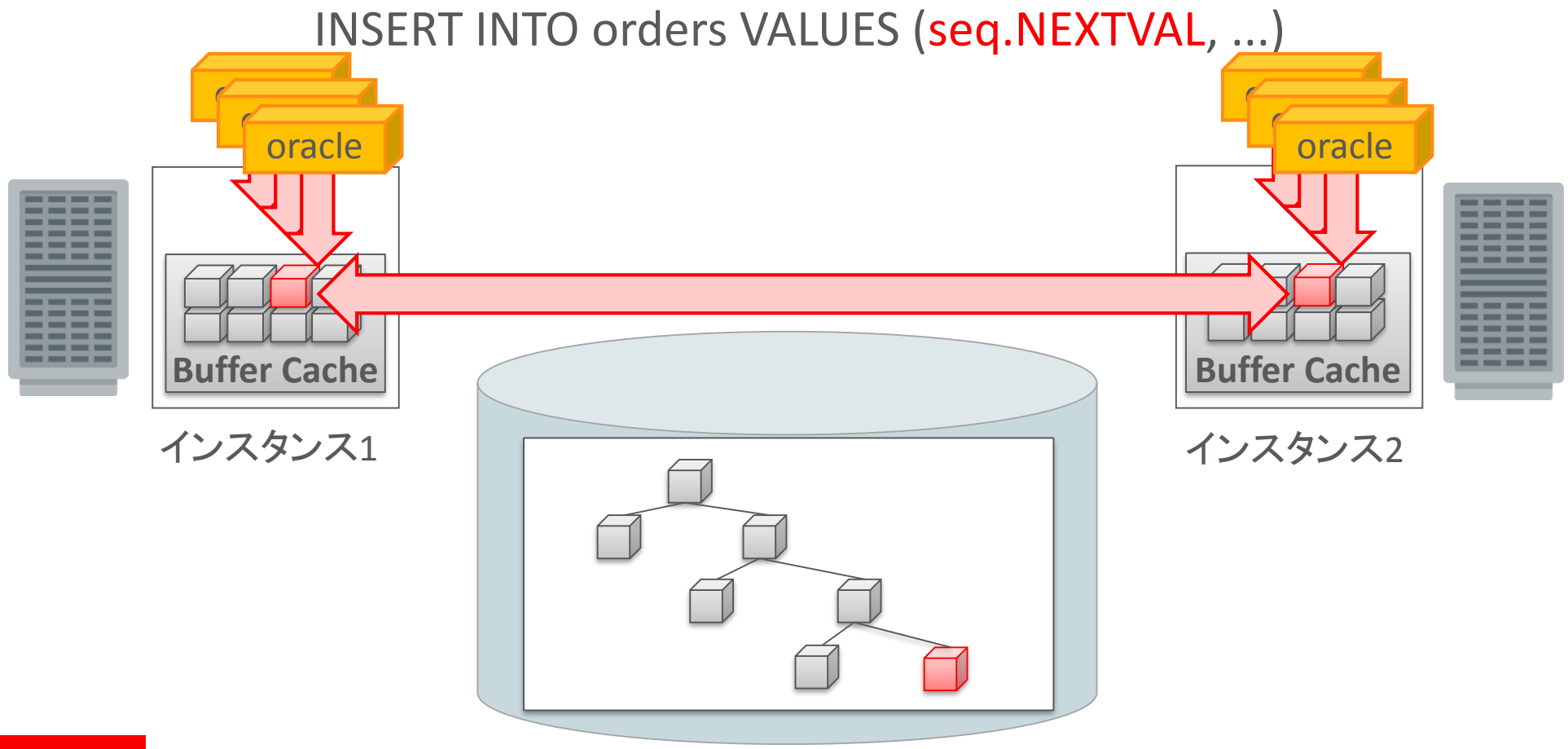
INSERT INTO orders VALUES (seq.NEXTVAL, ...)



同時セッション数が多い場合に更新競合

RACでの問題

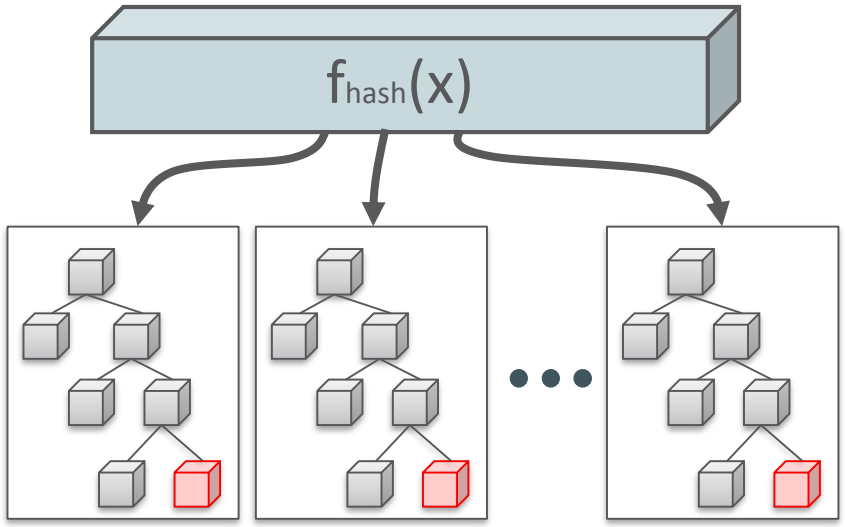
- 最も大きな値を格納するリーフ・ブロックの転送待ちが頻発する



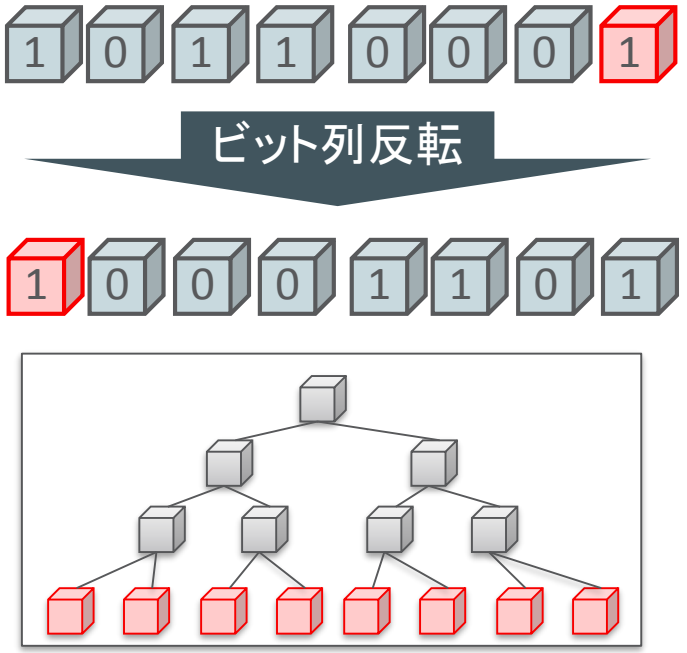
R12.2までの緩和方法

更新されるリーフ・ブロックを分散させる

INSERT INTO table_name VALUES (seq.NEXTVAL, ...)

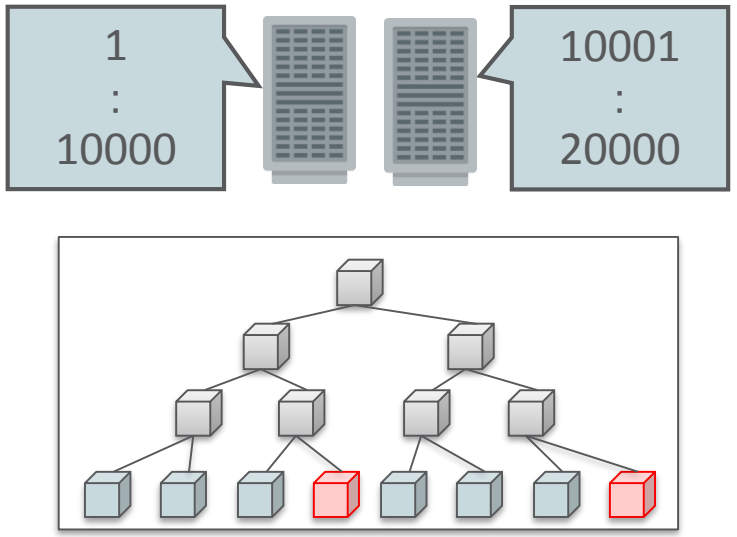


ハッシュ・パーティション索引



逆キー索引

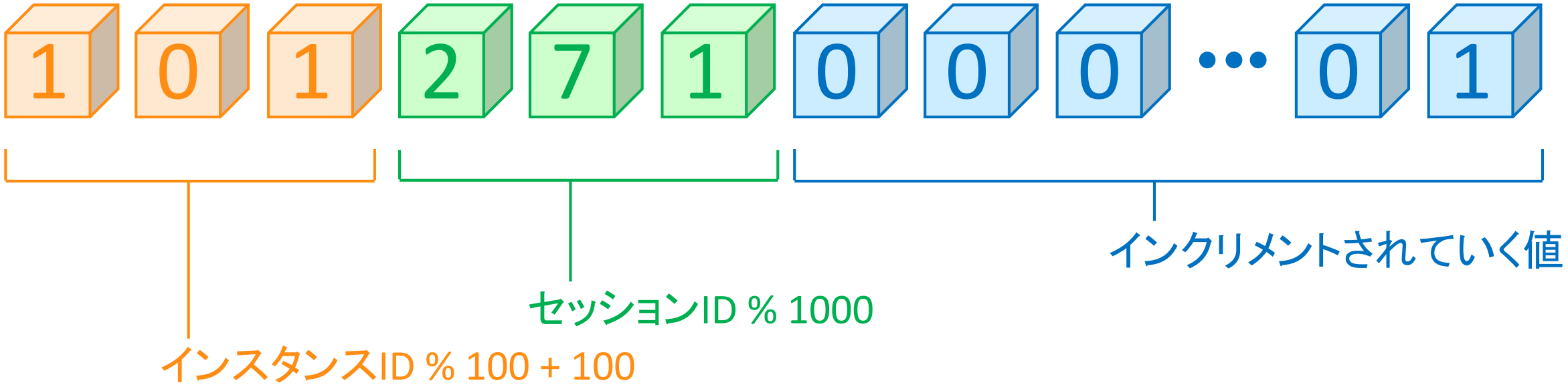
ALTER SEQUENCE seq
CACHE 10000 NOORDER;



CACHE値増加

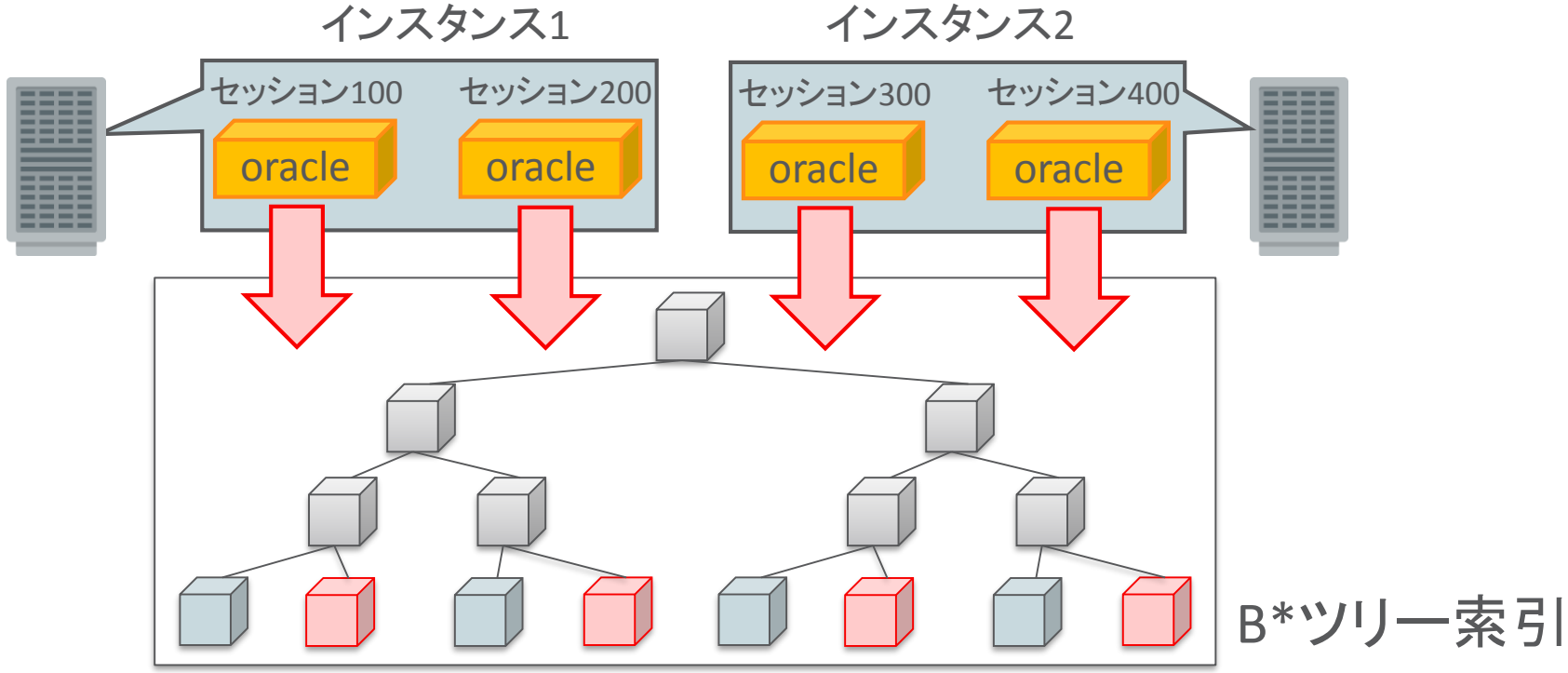
Scalable SEQUENCE

- CREATE SEQUENCE seq SCALE [NOEXTEND | EXTEND]
  - インスタンスIDとセッションIDがインクリメントされる値のプレフィックスに付く



Scalable SEQUENCE

- INSERT INTO table_name VALUES (seq.NEXTVAL, ...)
  - 異なるインスタンス間ではインスタンスIDで分散される
  - 同一インスタンス内の異なるセッション間ではセッションIDで分散される



CREATE SEQUENCE ... SCALE [NOEXTEND | EXTEND]

NOEXTENDとEXTENDの桁数の処理の違い

- NUMBER型 : 最大38桁
- （従来の）SEQUENCE : 最大28桁

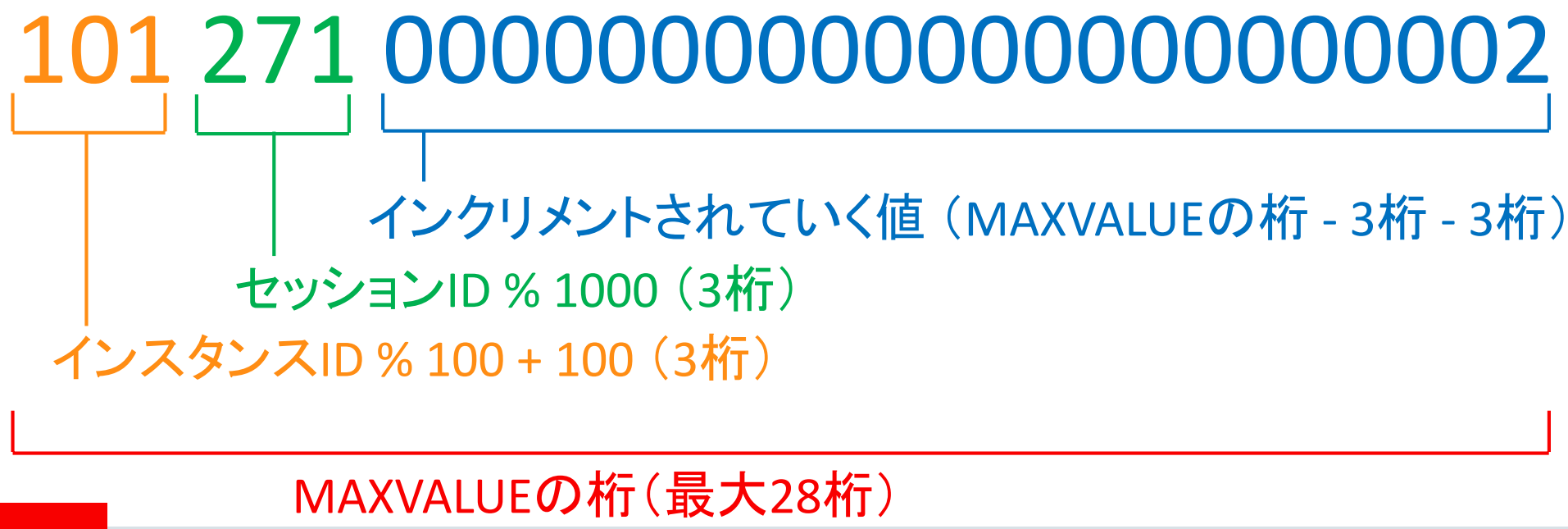
- SCALE NOEXTEND
  - プレフィックス6桁を含めて最大28桁

- SCALE EXTEND
  - プレフィックス6桁+最大28桁

CREATE SEQUENCE ... SCALE NOEXTEND;

従来のSEQUENCEの最大桁数に収まる

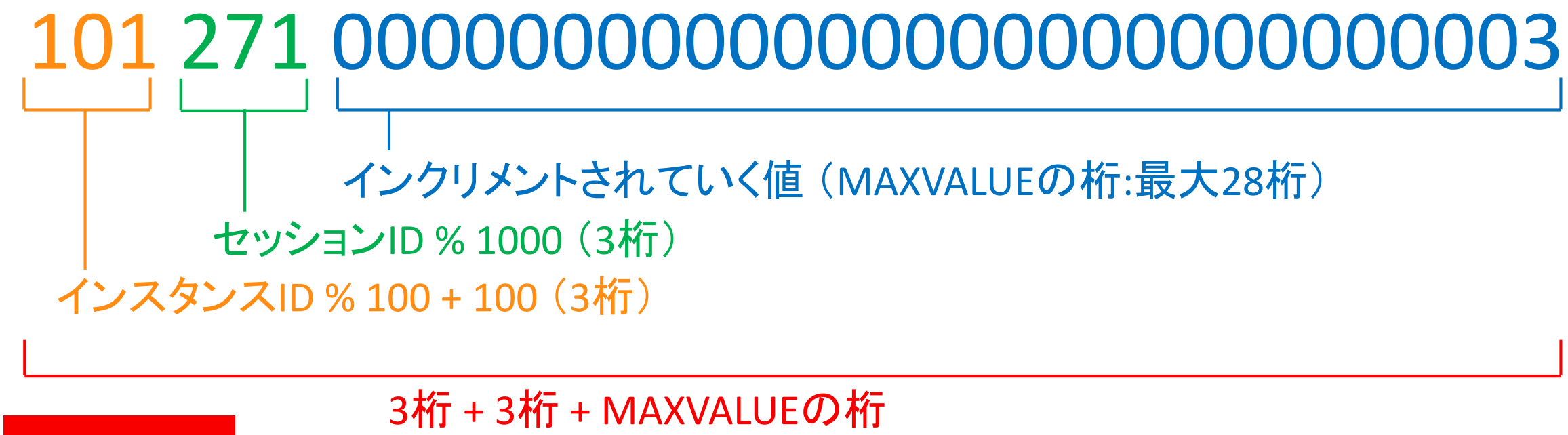
```
SQL> SELECT seq3.nextval as val FROM dual;
                                                  VAL
---------------------------------------------------
                  10127100000000000000000000002
```



CREATE SEQUENCE ... SCALE EXTEND;

SEQUENCEの最大桁数が拡張される（インクリメントされる桁の互換性）

```
SQL> SELECT seq2.nextval as val FROM dual;
                                                  VAL
---------------------------------------------------
          101271000000000000000000000000003
```



Scalable SEQUENCE

- 索引をつけた列にSEQUENCEの値をINSERTするときに発生する問題
  - 索引リーフ・ブロックの更新競合が発生しやすい
  - 更新ブロックを分散させて競合を緩和するチューニング

- Scalable SEQUENCE
  - 生成される値をインスタンスIDとセッションIDのプレフィックスで大きく離す
  - 連番でも単調増加でもない一意な値
  - 桁数が固定

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

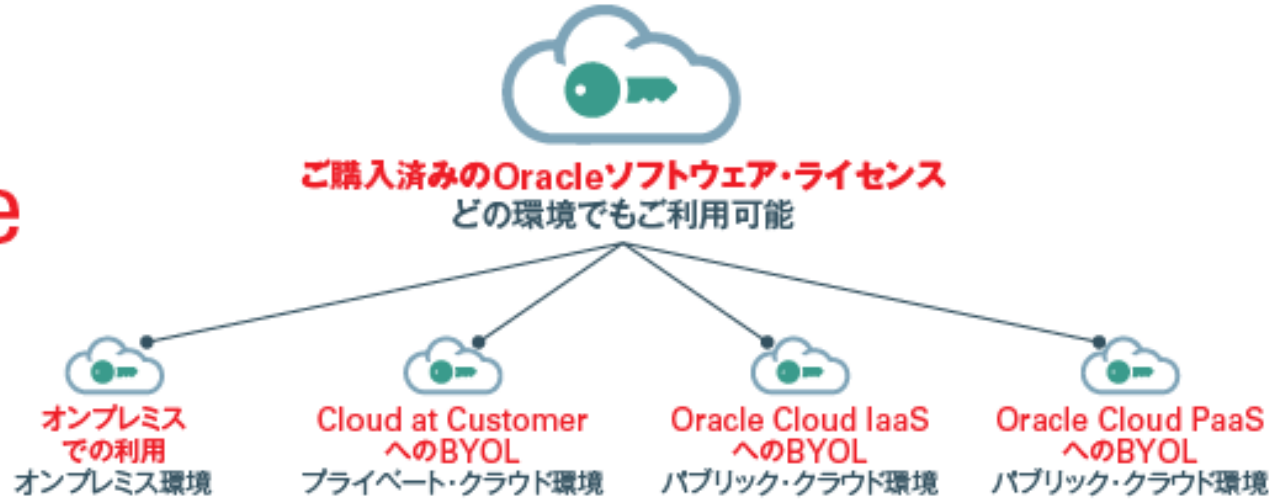
技術コラム しばちょう先生の試して納得！DBAへの道

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。

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ニュニケーションで、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。

Oracle Digital

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

ORACLE

Integrated Cloud

Applications & Platform Services


ORACLE®