

ORACLE®
NOSQL DATABASE

Oracle NoSQL Database – 親-子結合と集計

Oracle ホワイト・ペーパー | 2018 年 5 月



ORACLE®

目次

はじめに	1
親表-子表の結合	2
RDBMS LEFT OUTER JOIN との比較	3
例	4
集計	7
count(*)および count(any)	7
sum(any)	7
avg(any)	7
min (any)または max (any)	7
例	7
問合せ	8
まとめ	9

はじめに

Oracle NoSQL Database の表は、親/子の階層に分けて整理できます。作成可能な子表の数に制限はなく、子表がネスト可能な深さも制限はありません。データの取得において、デフォルトでは、親表に対して問合せを実行する場合、子表は取得されず、子表を取得する場合、親表は取得されません。これらの表はそれぞれ個別に扱われ、同じ階層の一部としてのデータ・リレーションシップを維持する手段をアプリケーションに提供します。

Oracle NoSQL Database における親/子表のリレーションシップは、従来のリレーションナル・データベースにおける親/子表のリレーションシップに非常に似ています。データの問合せに結合文が必要ない点が異なります。Oracle NoSQL Database が親/子表のデータを格納する方法のおかげで、このリレーションシップで維持されるデータの横断は非常に効率的です。親に関連付けられている子データセットは、全体が同じシャードに存在することを保証されており、1つのネットワーク・コールで取得できます。

Oracle NoSQL Database 18.1 より前のバージョンでは、親/子表から取得する場合、親表に問合せを実行しても子表のデータは取得されず、子表に問合せを実行しても親表のデータは取得されません。これらの表はそれぞれ別個に扱われ、表同士の結合を独立して扱うアプリケーションを必要としていました。

集計では、列の値に対して基本的な数値操作を行うことができます。これにより、データを分析に使用することができます。Oracle NoSQL Database 18.1 より前のバージョンでは、このような操作はクライアント・アプリケーションを使用するか、または MapReduce のバッチ・ジョブを経由する必要がありました。

Oracle NoSQL Database バージョン 18.1 は、こうした欠点に対処するため、2 つの便利な機能を採用しました。

- » 親-子表のリレーションを次のレベルへと進化させ、同じ表階層（親と子）に属する表の間での特別な種類の結合が追加されました。この結合は、一緒に配置されている行のみが互いに一致する可能性があるため、効率的に実行可能。
- » データに対し、sum、average、max などの集計演算を実行可能。

これらの機能については、下記で詳しく説明します。これら 2 つの機能の理解を助け、関連付けを行うため、以下の例では、ショッピング・カートを利用するアプリケーションにおける顧客とそのショッピング行動のユースケースで説明します。

親表-子表の結合

同じ階層に属する表から問合せを行うには、NESTED TABLES 句を使用する必要があります。構文は次のとおりです。

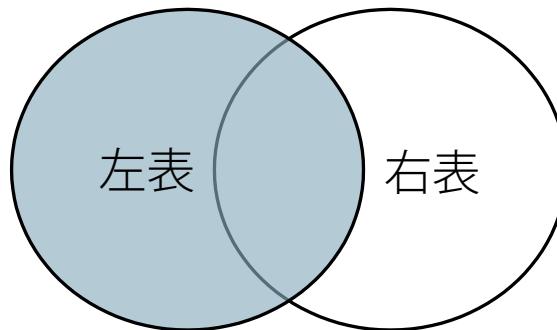
NESTED TABLES 句は、参加する表を指定し、3 つの独立したグループとして記述する必要があります。1 つめのグループはターゲット表を指定し、NESTED TABLES の後のかっこ内にインクルードされます。2 つめのグループは ANCESTORS（存在する場合）で、ターゲット表の後のかっこ内にインクルードされます。3 つめのグループは DESCENDANTS（存在する場合）で、ANCESTORS の後のかっこ内にインクルードされます。ANCESTORS および DESCENDANTS の個数は任意で、カンマで区切れます。ANCESTORS および DESCENDANTS は、ターゲット表に関連して行を抽出したい表を指定します。

構文は次のとおりです。

```
Nested Tables: from NESTED TABLES
  ( target_table
    ANCESTORS (ancestor1, ancestor2...)
    DESCENDANTS (descendant1, descendant2,...)
  )
```

RDBMS LEFT OUTER JOINとの比較

NESTED TABLES 句は、“Select columns from LeftTable LEFT JOIN RightTable on LeftTable.col1 = RightTable.col1”というサンプル構文を使用する、RDBMS スタイルの LEFT OUTER JOIN（左側外部結合）に似ています。LEFT OUTER JOIN を使用する問合せでは、“LEFT JOIN”キーワードの左側の表を左表、右側の表を右表と呼びます。問合せの結果には、右表内の行と一致する、左表内の行が含まれます。対応する行が右表にない行は、NULL を返します。



上のベン図は LEFT OUTER JOIN を示したもので、陰付きの部分が、返される行を表しています。2つの円が交差している部分は、両方の表で一致した行です。左側の円の残りの部分は、左表の中で右表の行と一致していない行です。したがって、左表のすべての行が結果に含まれます。

この結果は、ON 式が使用されている WHERE 句の部分でもあります。

NoSQL の留意事項：

- 問合せは、各シャードに対するパラレル・スキャン操作として実行され、高パフォーマンスで問合せが実行されます。
- ターゲット表に条件の一部として索引付き列が含まれる場合、索引は問合せで使用され、非常に高速に結果が取得されます。

これらは種類の異なる 2 つのデータベースであり、状況や要件が同じでもデータ管理のために交換して使用することはできません。

Oracle NoSQL Database では、子表は“.”を付ける表記法で表されます。

各データベースに表を作成することを読み手が知っていることが前提です。

例

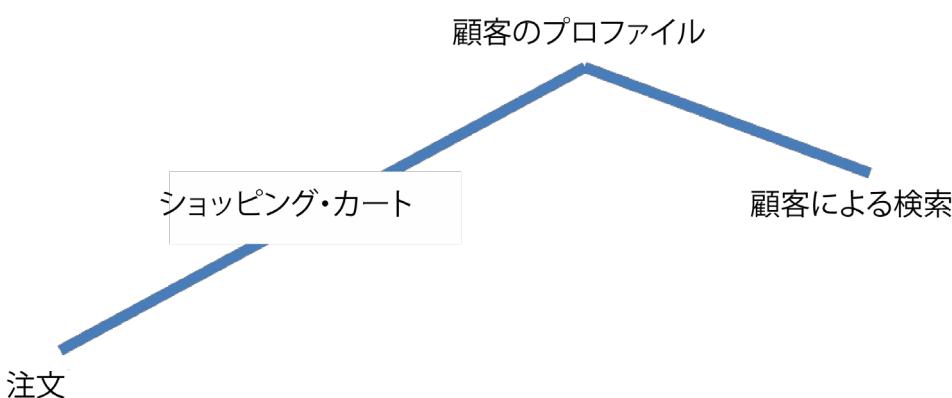
この新機能の効果を示す例として、Oracle NoSQL Database と MySQL（RDBMS スタイル）との比較で、同じような表から得られた結果を示します。

表

MySQL – CustomerProfile、Cart、Order、Searched

Oracle NoSQL Database – customerProfile、customerProfile.cart、customerProfile.cart.order、customerProfile.searched

Oracle NoSQL Database における表階層を図で示すと次のようになります。



これに基づき、上記の表から単純な抽出を行う問合せは次のようなものになります。

-----Mike が検索し、注文したすべての商品。サイトでの各注文のステータス-----

MySQL と Oracle NoSQL Database の両方について、表スクリプト、挿入文および問合せが zip ファイルとして提供されています。

MySQL - MySQL データベースでの問合せは次のようになります。

```
select * from customer left outer join cart on
customer.id = cart.customer_id left outer join searched
on customer.id = searched.customer_id left outer join
sys.order on cart.cart_id=sys.order.cart_id;
```

出力は次のとおりです。簡潔にするためごく一部を示します。

```
{"id": 1, "name": "Mike", "lastname": "Brey", "phone_no": "8888888888", "cart_id": 1001, "customer_id": 1, "product_Info": {"Date": {"ExpiryDate": "May 2019", "PurchaseDate": "May 2017"}, "Price": 30, "Quantity": 1, "ProductID": 1101, "Description": "Table", "ProductName": "Table"}, "searched_id": 10, "customer_id": 1, "product_Info": {"Date": {"ExpiryDate": "May 2019", "PurchaseDate": "May 2017"}, "Price": 30, "Quantity": 1, "ProductID": 1101, "Description": "Table", "ProductName": "Table"}, "order_id": 1101, "cart_id": 1001, "status": "paid"}
```

```
{"id" : 1,"name" :"Mike","lastname" :"Brey","phone_no" :"8888888888","cart_id" : 1002, "customer_id" : 1, "product_Info" : {"\Date": {"\ExpiryDate": "May 2019", "\PurchaseDate": "June 2017"}, "\Price": 110, "\Quantity": 1, "\ProductID": 1102, "\Description": "Teak Table", "\ProductName": "Teak Table"}, "searched_id": 11, "customer_id": 1, "product_Info": {"\Date": {"\ExpiryDate": "Dec 2019"}, "\PurchaseDate": "Jan 2017"}, "\Price": 1200, "\Quantity": 1, "\ProductID": 1201, "\Description": "TV", "\ProductName": "TV"}, "order_id": 1101, "cart_id": 1001, "status": "cod"}
```

```
{"id" : 1,"name" :"Mike","lastname" :"Brey","phone_no" :"8888888888","cart_id" : 1003, "customer_id" : 1, "product_Info" : {"\Date": {"\ExpiryDate": "Dec 2022"}, "\PurchaseDate": "Jan 2017"}, "\Price": 350, "\Quantity": 1, "\ProductID": 1103, "\Description": "Dining Table", "\ProductName": "Dining Table"}, "searched_id": 12, "customer_id": 1, "product_Info": {"\Date": {"\ExpiryDate": "Jan 2020"}, "\PurchaseDate": "Jan 2017"}, "\Price": 3400, "\Quantity": 1, "\ProductID": 1301, "\Description": "Refrigirator", "\ProductName": "Refrigirator"}, "order_id": 1101, "cart_id": 1001, "status": "paid"}
```

```
{"id" : 1,"name" :"Mike","lastname" :"Brey","phone_no" :"8888888888","cart_id" : 1004, "customer_id" : 1, "product_Info" : {"\Date": {"\ExpiryDate": "May 2019"}, "\PurchaseDate": "June 2017"}, "\Price": 35, "\Quantity": 6, "\ProductID": 1103, "\Description": "Chairs", "\ProductName": "Chairs"}, "searched_id": 13, "customer_id": 1, "product_Info": {"\Date": {"\ExpiryDate": "Dec 2020"}, "\PurchaseDate": "Sep 2017"}, "\Price": 1800, "\Quantity": 1, "\ProductID": 1401, "\Description": "Dish Washer", "\ProductName": "Dish Washer"}, "order_id": 1101, "cart_id": 1001, "status": "paid"}}
```

.....

Oracle NoSQL Database - Oracle NoSQL Database における同様の問合せは次のようになります。

```
select * from NESTED TABLES (customerProfile cp
DESCENDANTS (customerProfile.cart cc,
customerProfile.cart.order,customerProfile.searched))
```

出力は次のとおりです。全体で 13 行が返されますが、簡潔にするため一部のみ示します。

```
{"cp":{"id":1,"name":"Mike","lastname":"Brey","phone_no":"8888888888"},"cc":{"id":1,"cart_id":1001,"customer_id":1,"product_Info":{"Date":{"ExpiryDate":"May 2019","PurchaseDate":"May 2017"},"Description":"Table","Price":30,"ProductID":1101,"ProductName":"Table","Quantity":1}),"customerProfile_cart_order":{"id":1,"cart_id":1001,"co_id":1101,"status":"paid"},"customerProfile_searched":null}

{"cp":{"id":1,"name":"Mike","lastname":"Brey","phone_no":"8888888888"},"cc":{"id":1,"cart_id":1002,"customer_id":1,"product_Info":{"Date":{"ExpiryDate":"May 2019","PurchaseDate":"June 2017"},"Description":"Teak Table","Price":110,"ProductID":1102,"ProductName":"Teak Table","Quantity":1}),"customerProfile_cart_order":{"id":1,"cart_id":1002,"co_id":1102,"status":"cod"},"customerProfile_searched":null}

{"cp":{"id":1,"name":"Mike","lastname":"Brey","phone_no":"8888888888"},"cc":{"id":1,"cart_id":1003,"customer_id":1,"product_Info":{"Date":{"ExpiryDate":"Dec 2022","PurchaseDate":"Jan 2017"},"Description":"Dining Table","Price":350,"ProductID":1103,"ProductName":"Dining Table","Quantity":1}),"customerProfile_cart_order":{"id":1,"cart_id":1003,"co_id":1103,"status":"paid"},"customerProfile_searched":null}

{"cp":{"id":1,"name":"Mike","lastname":"Brey","phone_no":"8888888888"},"cc":{"id":1,"cart_id":1004,"customer_id":1,"product_Info":{"Date":{"ExpiryDate":"May 2019","PurchaseDate":"June 2017"},"Description":"Chairs","Price":35,"ProductID":1103,"ProductName":"Chairs","Quantity":6}),"customerProfile_cart_order":{"id":1,"cart_id":1004,"co_id":1104,"status":"cod"},"customerProfile_searched":null}

{"cp":{"id":1,"name":"Mike","lastname":"Brey","phone_no":"8888888888"},"cc":null,"customerProfile_cart_order":null,"customerProfile_searched":{"id":1,"cs_id":10,"product_Info":{"Date":{"ExpiryDate":"May 2019","PurchaseDate":"May 2017"},"Description":"Table","Price":30,"ProductID":1101,"ProductName":"Table","Quantity":1}}}

{"cp":{"id":1,"name":"Mike","lastname":"Brey","phone_no":"8888888888"},"cc":null,"customerProfile_cart_order":null,"customerProfile_searched":{"id":1,"cs_id":11,"product_Info":{"Date":{"ExpiryDate":"Jan 2020","PurchaseDate":"Jan 2017"},"Description":"Refrigerator","Price":3400,"ProductID":1201,"ProductName":"Refrigerator","Quantity":1}}}

{"cp":{"id":1,"name":"Mike","lastname":"Brey","phone_no":"8888888888"},"cc":null,"customerProfile_cart_order":null,"customerProfile_searched":{"id":1,"cs_id":12,"product_Info":{"Date":{"ExpiryDate":"Dec 2019","PurchaseDate":"June 2017"},"Description":"Microwave Owen","Price":800,"ProductID":1301,"ProductName":"Microwave Owen","Quantity":1}}}

.....
```

集計

Oracle NoSQL Database の集計関数は行ごとに式を評価し、それぞれの戻り値を集計して单一の値にします。構文的に集計関数は SELECT 句に現れます。SELECT 句に集計関数が含まれる場合、FROM 句または WHERE 句によって生成される一連の行全体が対象となり、この単一のグループについて集計関数が評価されます。現在利用可能な集計関数は、count、sum、avg、min、max です。

count(*)およびcount(any)

グループに含まれる行数を返します。または、グループの各行について入力式を計算し、NULL 値以外のすべての値の個数を返します。

sum(any)

各行について入力式を計算し、シーケンスに含まれるすべての数値を合算します。結果の値のタイプは、入力値に基づきます。整数タイプの入力項目が 1 つある場合、結果は整数になります。double または float のタイプの入力が 1 つある場合、結果は double で返されます。

注：戻り値タイプが double の場合に、2 個の浮動小数点値を合計しようとすると、既知の Java エラーが発生します。

avg(any)

グループの各行について入力式を計算します。シーケンス内のすべての数値の合計と個数を計算します。結果の値は average (合計 ÷ 個数) で、タイプは入力値に基づきます。整数タイプの入力項目が 1 つある場合、結果は整数になります。double または float のタイプの入力が 1 つある場合、結果の値は double で返されます。

min (any)またはmax (any)

グループに含まれる各行について入力式を計算し、そのグループの式の最小値または最大値を返します。

例

前述のショッピング・カートの例を用い、以下を求めます。

1. ある特定の顧客が検索した製品の総数。
2. ある日付までに顧客が行った注文の合計数。
3. ある日付までに顧客が購入した最高金額。
4. ある日付までに顧客が購入した最小金額。
5. ある顧客の平均購入額

問合せ

- customerProfile 表において、Mike は id が 1 です。Mike は、customerProfile.searched 表ではこの id で参照されます。Mike が実行した検索の回数の問合せは次のようにになります。

```
sql -> select count(*) from customerProfile.searched where
id=1;
{"column_1":4}

1 row returned
```

- customerProfile 表において、Mike は id が 1 です。Mike は、customerProfile.cart 表ではこの id で参照されます。Mike が購入した合計金額の問合せは次のようにになります。

```
sql -> Select sum(cc.product_info.Price) from
customerProfile.cart cc where id=1;
{"column_1":525}

1 row returned
```

- customerProfile 表において、Mike は id が 1 です。Mike は、customerProfile.cart 表ではこの id で参照されます。Mike が購入したもっとも高額なものの問合せは次のようにになります。

```
sql -> Select max(cc.product_info.Price) from
customerProfile.cart cc where id=1;
{"column_1":350 }

1 row returned
```

- customerProfile 表において、Mike は id が 1 です。Mike は、customerProfile.cart 表ではこの id で参照されます。Mike が購入したもっとも低額なものの問合せは次のようにになります。

```
sql -> Select min(cc.product_info.Price) from
customerProfile.cart cc where id=1;
{"column_1":30}

1 row returned
```

- customerProfile 表において、Mike は id が 1 です。Mike は、customerProfile.cart 表ではこの id で参照されます。Mike が購入で費やした平均額の問合せは次のようにになります。

```
sql -> Select avg(cc.product_info.Price) from
customerProfile.cart cc where id=1;
{"column_1":131.25}

1 row returned
```

まとめ

強化された問合せ機能を持つ Oracle NoSQL Database により、親-子表として示される表で構成された同一階層に格納されている関連データの取得が容易になります。問合せ機能に集計を含めることで、顧客が格納したデータからの有効なデータの取得も非常に容易になります。



CONNECT WITH US

- blogs.oracle.com/oracle
- facebook.com/oracle
- twitter.com/oracle
- oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2018, 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默示的保証を含め、商品性ないし特定目的適合性に関する默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画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。

Oracle NoSQL Database 18.1 の新機能 - 親-子結合と集計 2018 年 5 月
著者 : Vishal Settipalli
共著者 : Michael Schulman, Michael Brey, Tim Goh

| Oracle is committed to developing practices and products that help protect the environment