



VERTICA

10.1 新機能 紹介

マイクロフォーカスエンタープライズ株式会社
2021年3月

テーブル

- テーブルのカラム数上限の変更
 - テーブルは最大9,800のカラムまで定義できるように拡張。
- Flattened TableのSET USING/DEFAULT句の拡張
 - サブクエリを用いて他のテーブルから値を取得し、その値を同じSQL式の中で、Flattened Tableのカラムの値と一緒に使用できるように拡張。

```
=> ALTER TABLE customer_orders ADD COLUMN gmt_delivery_time timestamp
      DEFAULT NEW_TIME(customer_orders.local_delivery_time,
                       (SELECT c.customer_tz FROM customers c
                        WHERE (c.customer_key = customer_orders.customer_key)),
                       'GMT');
```

- 以前は定義できずに以下のエラーが発生。

```
ROLLBACK 7421: Expressions on SELECT statements cannot be used in default query definitions
```

データ型

- Parquetファイルを使ったExternal TableにおけるStruct・Array型の複合型のサポート
 - ParquetファイルはStruct型（VerticaではROW型）、配列/Array型（VerticaではARRAY型）を持つことができる。
 - Parquetファイルを使ったExternal Tableで、Array型のフィールドを持つStruct型、Struct型の配列、などの複合型を定義することが可能。
 - 最大100階層までネストさせることが可能。

```
=> CREATE EXTERNAL TABLE customers (  
  name VARCHAR,  
  contact ROW (  
    street VARCHAR,  
    city VARCHAR,  
    zipcode INT,  
    email ARRAY[VARCHAR]  
  ),  
  accountid INT  
)  
AS COPY FROM '...' PARQUET;
```

データ型

- ORCファイルでのStruct型のサポート
 - ORCファイルを使ったExternal Tableで、ROW型を用いてORCのStruct型を扱うことが可能。
 - 配列/Array型とMap型はExternal Tableで定義は可能だがデータの取得はできない。

```
=> CREATE EXTERNAL TABLE customers (  
    name VARCHAR,  
    address ROW(street VARCHAR, city VARCHAR, zipcode INT),  
    accountID INT  
)  
AS COPY FROM '...' ORC;
```

データロード

- Parquetファイルのメタデータをキャッシュ
 - 実行計画を作成する際、Parquetファイルのメタデータをキャッシュし、実行時に再利用する。
 - 読み込み回数の削減、ネットワーク処理による遅延の削減に効果あり。
 - ファイルの読み込みにAPIを使用するケースにおいてはコスト削減の効果もあり。
 - キャッシュはTEMP Storage Locationを使用
- JSON, Avroパーサーによる配列のサポート
 - JSONおよびAvroのデータをロードする際、一次元の配列を直接ロードすることが可能。
 - 多次元配列のデータは引き続きflatten_mapsとflatten_arraysのパラメータを使用してロード。

```
=> CREATE EXTERNAL TABLE customers(id INT, address VARCHAR, transactions ARRAY[INT])  
AS COPY FROM '...' PARSER fjsonparser();
```

データロード

- Parquetパーサーによる緩いスキーママッチングのサポート
 - デフォルトではパーサーは厳しいスキーママッチングを使用。Parquetデータ内のすべてのカラムを定義された順番通りに取得する必要がある。
 - 緩いスキーママッチングをサポート。Parquetデータ内の必要なカラムのみを取得できる。ただし、カラム名およびデータ型を一致させる必要がある。
 - 緩いスキーママッチングで使用できるデータ型は基本型と基本型を持つ一次元の配列のみ。

```
=> CREATE EXTERNAL TABLE restaurants(name VARCHAR, cuisine VARCHAR)
AS COPY FROM '/data/rest*.parquet'
PARQUET(allow_long_varbinary_match_complex_type='True',
        do_soft_schema_match_by_name='True');
```

SQLファンクションおよびステートメント

- CREATE OR REPLACE VIEWに必要な権限
 - Viewを作成するスキーマに対するCREATE権限とそのViewに対するDROP権限を持つユーザはCREATE OR REPLACE VIEWを使ってViewの置き換えが可能。
 - 以前はViewのOwnerのみがCREATE OR REPLACE VIEWを使用できた。
- IMplodeファンクション
 - GROUP BYによってまとめられたカラムの値を配列として生成。

```
=> SELECT * FROM filtered;
```

position	itemprice	itemkey
0	14.99	345
0	27.99	567
1	18.99	567
1	35.99	345
2	14.99	123

```
=> SELECT itemkey, IMplode(itemprice) AS prices  
FROM filtered  
GROUP BY itemkey ORDER BY itemkey;
```

itemkey	prices
123	[14.99]
345	[14.99, 35.99]
567	[27.99, 18.99]

SQLファンクションおよびステートメント

- EXPLODEファンクションの拡張

- 一度のEXPLODEファンクションの呼び出しで複数の配列を展開させることが可能。
- 以前は一度に1つの配列のみ展開できたため、複数の配列を展開させる場合はサブクエリをネストさせて対応する必要があった。

```
=> SELECT orderkey, custkey, prodkey, orderprices, email_addrs FROM orders LIMIT 5;
```

orderkey	custkey	prodkey	orderprices
113-341987	342799	["MG-7190 ", "VA-4028 ", "EH-1247 ", "MS-7018 "]	["60.00", "67.00", "22.00", "14.99"]
111-952000	342845	["ID-2586 ", "IC-9010 ", "MH-2401 ", "JC-1905 "]	["22.00", "35.00", null, "12.00"]
111-345634	342536	["RS-0731 ", "SJ-2021 "]	["50.00", null]
113-965086	342176	["GW-1808 "]	["108.00"]
111-335121	342321	["TF-3556 "]	["50.00"]

SQLファンクションおよびステートメント

- EXPLODEファンクションの拡張 (続き)

```
=> SELECT EXPLODE(orderkey, prodkey, orderprices USING PARAMETERS explode_count=2)
      OVER(PARTITION BEST)
      AS (orderkey, pk_idx, pk_val, ord_idx, ord_val)
      FROM orders
      WHERE orderkey='113-341987';
```

orderkey	pk_idx	pk_val	ord_idx	ord_val
113-341987	0	MG-7190	0	60.00
113-341987	0	MG-7190	1	67.00
113-341987	0	MG-7190	2	22.00
113-341987	0	MG-7190	3	14.99
113-341987	1	VA-4028	0	60.00
113-341987	1	VA-4028	1	67.00
113-341987	1	VA-4028	2	22.00
113-341987	1	VA-4028	3	14.99
113-341987	2	EH-1247	0	60.00
113-341987	2	EH-1247	1	67.00
113-341987	2	EH-1247	2	22.00
113-341987	2	EH-1247	3	14.99
...				

データベース管理

- ノードのヘルスチェックの変更
 - ノードのヘルスチェックの際、Verticaプロセスが次のStorage Locationに対し読み書き可能かを確認。
 - ・ TEMP, DATA
 - ・ TEMP
 - ・ DATA
 - ・ DEPOT

バックアップ・リストア

- Eon Modeでのオブジェクトレベルのバックアップ・リストア
- バックアップ先としてのGoogle Cloud Storage
 - Google Cloud Platform上で動作するEnterprise ModeとEon Modeの両方のデータベースで、バックアップ先としてGoogle Cloud Storageの指定が可能。
- vbr設定ファイルの変更
 - vbr設定ファイルの[S3]セクションが[CloudStorage]に変更。
 - パラメータ名の接頭辞も「s3_」から「cloud_storage_」に変更。

[10.0まで]

s3_backup_file_system_path
s3_backup_path
s3_encrypt_transport
s3_concurrency_backup
s3_concurrency_restore
s3_encrypt_at_rest
s3_sse_kms_key_id

[10.1から]

cloud_storage_backup_file_system_path
cloud_storage_backup_path
cloud_storage_encrypt_transport
cloud_storage_concurrency_backup
cloud_storage_concurrency_restore
cloud_storage_encrypt_at_rest
cloud_storage_sse_kms_key_id

バックアップ・リストア

- 新しいvbr設定パラメータ
 - [CloudStorage]セクション
 - cloud_storage_ca_bundle
 - cloud_storage_concurrency_delete
 - [Transmission]セクション
 - concurrency_delete
- 新しいvbr設定ファイルサンプル
 - backup_restore_cloud_storage.ini
 - ・ Enterprise ModeとEon ModeでCloud Storageを用いたバックアップ・リストアの設定サンプル
 - ・ 以前のbackup_restore_S3.iniとeon_backup_restore.iniと入れ替え

セキュリティ・ユーザ認証

- デフォルトのパスワードハッシュアルゴリズムの変更
 - パスワードのMD5によるハッシュは非推奨となり今後のリリースで削除予定。
 - SecurityAlgorithmパラメータのデフォルト値はSHA512に変更（以前はMD5）。
 - SecurityAlgorithmパラメータ値 NONE は削除（以前はNONE = MD5）。
 - 既存のデータベースをアップグレードした場合、SecurityAlgorithmパラメータ値がMD5の場合は変更なし。NONEの場合はMD5に変更される。
- 新しいPROFILEのパラメータ
 - 次のパラメータがパスワードプロファイルの設定に追加。
 - ・ PASSWORD_MIN_CHAR_CHANGE
新しいパスワードが古いパスワードと最低限違わなければならない文字数。
 - ・ PASSWORD_MIN_LIFE_TIME
パスワードを変更できるようになるまでの日数。

セキュリティ・ユーザ認証

- 事前にハッシュされたパスワードのユーザへの付与
 - CREATE USERおよびALTER USERで事前にハッシュされたパスワードをサポート。

```
=> CREATE USER Alice IDENTIFIED BY  
    'sha512e0299de83ecfaa0b6c9cbb1feabfbe0b3c82a1495875cd9ec1c4b09016f09b42c1'  
    SALT '465a4aec38a85d6ecea5a0ac8f2d36d8' ;
```

Eon Mode

- Depotへのピン留めの拡張
 - Projection単位でDepotにピン留めすることが可能。
 - SET_DEPOT_PIN_POLICY_PROJECTION
 - CLEAR_DEPOT_PIN_POLICY_PROJECTION
 - SET_DEPOT_PIN_POLICY_PARTITION (既存のファンクションを拡張)
- Subclusterの複製
 - admintoolsのdb_add_subclusterを使用して既存するSubclusterの複製を作成することが可能。
 - ほとんどの設定を継承して新しいSubclusterを作成。

```
$ admintools -t db_add_subcluster -d verticadb ¥  
-s 10.11.12.13, 10.11.12.14, 10.11.12.15 ¥  
-p mypassword --like=analytics_1 -c analytics_2
```

Management Console

- パスワードのセキュリティ強化
 - MCユーザのパスワードに次の要件が追加。
 - ・ Minimum password length
 - ・ Password minimum lifetime / Password lifetime
 - ・ Login attempt limit
 - ・ Account lockout duration
 - ・ Security Questions

Apache Kafka Integration

- Notifierの暗号化
 - Notifierの暗号化が次のパラメータを用いて可能。
 - TLSMODE
 - CA BUNDLE
 - CERTIFICATE

User-Defined Extensionsの拡張

- Scalar Functionの拡張
 - User-Defined Scalar Function (UDSF) が複数の値をROW型を用いて返すことをサポート。
- Functionの引数の上限を拡張
 - User-Defined Functionの引数の上限が9,800まで拡張。



VERTICA

www.vertica.com

www.microfocus-enterprise.co.jp