MySQL 8.0 の薄い本【追補版】

hmatsu47　著

2020-07-23 MySQL 8.0.21 版　　発行

付録 A

MySQL 8.0.21 の変更と新機能

MySQL 8.0.21 の変更と新機能について、主なものをピックアップしました。

A.1 MySQL 8.0 のインストールと設定パラメータ（第 1 章）

A.1.1 管理用クライアント専用のネットワーク設定が可能に（一般クライアント設定と分離）

公式リファレンスマニュアル

- https://dev.mysql.com/doc/refman/8.0/en/administrative-connection-interface.html#administrative-interface-encrypted-connections
- https://dev.mysql.com/doc/refman/8.0/en/performance-schema-tls-channel-status-table.html

A.2 アカウント管理（第 2 章）

A.2.1 CREATE USER・ALTER USER で JSON 形式のユーザコメントが登録可能に

公式リファレンスマニュアル

- https://dev.mysql.com/doc/refman/8.0/en/create-user.html
  - 「Beginning with MySQL 8.0.21, …」
- https://dev.mysql.com/doc/refman/8.0/en/alter-user.html
  - 「MySQL 8.0.21 and later supports user comments and user attributes, …」

A.3 JSON ドキュメントストアの新機能（第 5 章）

A.3.1 JSON_VALUE() 関数

JSON ドキュメントからの値の取り出しが少しだけ楽になりました。

公式リファレンスマニュアル

- https://dev.mysql.com/doc/refman/8.0/en/json-search-functions.html#function_json-value

ブログ記事等

- https://elephantdolphin.blogspot.com/2020/07/jsonvalue-now-in-mysql-8021.html
- 著者ブログ
  - https://qiita.com/hmatsu47/items/013da2971c8934d867e7

実行例

関数インデックスとして使ってみます。

```
mysql> CREATE DATABASE jsontest;
Query OK, 1 row affected (0.01 sec)

mysql> USE jsontest;
Database changed
mysql> CREATE TABLE t1(
    ->   j JSON,
    ->   INDEX i1 ( (JSON_VALUE(j, '$.id' RETURNING UNSIGNED)) )
    -> );
Query OK, 0 row affected (0.02 sec)
```

※ここでデータを投入。

```
mysql> SELECT * FROM t1;
+----------------------------------+
| j                                |
+----------------------------------+
| {"id": 100, "val": [1, 2, 3]}    |
| {"id": 101, "val": [4, 5, 6, 7]} |
| {"id": 110, "val": [8, 9, 0]}    |
| {"id": 120, "val": [1, 2]}       |
| {"id": 122, "val": 3}            |
| {"id": 130, "val": [4, 5]}       |
| {"id": 140, "val": [6, 7, 8]}    |
| {"id": 150, "val": [9, 0, 1, 2]} |
| {"id": 200, "val": [3, 4, 5]}    |
| {"id": 220, "val": [6, 7]}       |
+----------------------------------+
10 rows in set (0.00 sec)

mysql> EXPLAIN SELECT * FROM t1 WHERE JSON_VALUE(j, '$.id' RETURNING UNSIGNED) = 150\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: t1
   partitions: NULL
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 10
     filtered: 100.00
        Extra: Using where
```

```
1 row in set, 1 warning (0.00 sec)
```

※インデックスが使われなかったのでテーブル定義を確認。

```
mysql> SHOW CREATE TABLE t1\G
*************************** 1. row ***************************
       Table: t1
Create Table: CREATE TABLE `t1` (
  `j` json DEFAULT NULL,
  KEY `i1` ((json_value(`j`, _utf8mb4'$.id' returning unsigned)))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
1 row in set (0.00 sec)
```

※「_utf8mb4」が補完されていた。

```
mysql> EXPLAIN SELECT * FROM t1 WHERE json_value(`j`, _utf8mb4'$.id' returning unsigned) = 150
\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: t1
   partitions: NULL
         type: ref
possible_keys: i1
          key: i1
      key_len: 9
          ref: const
         rows: 1
     filtered: 100.00
        Extra: NULL
1 row in set, 1 warning (0.00 sec)
```

※インデックスの定義に合わせて WHERE 句を書いてみたところ、うまく動いた。

A.4　レプリケーションの新機能（第 7 章）

A.4.1　グループレプリケーションの強化

主に安定性・可用性向上を目的とするグループレプリケーションの機能強化が行われました。

ブログ記事等

- https://mysqlhighavailability.com/mysql-8-0-21-replication-enhancements/
- https://mysqlhighavailability.com/you-can-now-use-binary-log-checksums-with-group-replication/
  - グループレプリケーションでバイナリログのチェックサムをサポート
- https://mysqlhighavailability.com/mysql-group-replication-default-response-to-network-partitions-has-changed/
  - 可用性向上のために 2 つのシステム変数のデフォルト値を変更

A.4.2　MySQL Shell の新機能

ブログ記事等

- https://mysqlserverteam.com/whats-new-in-mysql-shell-8-0-21/
  - 論理ダンプ・リストアツールの導入など

A.5　オプティマイザと InnoDB の新機能（第 8 章）

A.5.1　オプティマイザスイッチ

`subquery_to_derived`・`prefer_ordering_index` が追加されました。

公式リファレンスマニュアル

- https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_optimizer_switch
- https://dev.mysql.com/doc/refman/8.0/en/switchable-optimizations.html

A.5.2　UPDATE・DELETE でセミジョイン（準結合）・マテリアライズ（実体化）最適化をサポート

サブクエリを使った更新・削除が高速化されました。

ブログ記事等

- 著者ブログ
  - https://qiita.com/hmatsu47/items/aa687aa30c4570bac861

実行例

UPDATE で試してみます。

```
mysql> USE multitable_test;
Database changed
```

※テーブル構造は以下の通り。

```
mysql> SHOW CREATE TABLE t1\G
*************************** 1. row ***************************
       Table: t1
Create Table: CREATE TABLE `t1` (
  `id` int NOT NULL AUTO_INCREMENT,
  `key` int NOT NULL,
  PRIMARY KEY (`id`),
  KEY `key` (`key`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
1 row in set (0.00 sec)
```

```
mysql> SHOW CREATE TABLE t2\G
*************************** 1. row ***************************
       Table: t2
Create Table: CREATE TABLE `t2` (
  `id` int NOT NULL AUTO_INCREMENT,
  `val` int NOT NULL,
  `key` int NOT NULL,
  PRIMARY KEY (`id`),
  KEY `valkey` (`val`,`key`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
1 row in set (0.00 sec)
```

※それぞれ 50 行のデータを投入。

```
mysql> EXPLAIN UPDATE t1 SET t1.key = 0 WHERE t1.key IN (SELECT t2.key FROM t2 WHERE t2.val IN
(40, 60, 100))\G
*************************** 1. row ***************************
           id: 1
  select_type: UPDATE
        table: t1
   partitions: NULL
         type: ALL
possible_keys: key
          key: NULL
      key_len: NULL
          ref: NULL
         rows: 44
     filtered: 100.00
        Extra: Using where
*************************** 2. row ***************************
           id: 1
  select_type: SIMPLE
        table: <subquery2>
   partitions: NULL
         type: eq_ref
possible_keys: <auto_distinct_key>
          key: <auto_distinct_key>
      key_len: 4
          ref: multitable_test.t1.key
         rows: 1
     filtered: 100.00
        Extra: NULL
*************************** 3. row ***************************
           id: 2
  select_type: MATERIALIZED
        table: t2
   partitions: NULL
         type: range
possible_keys: valkey
          key: valkey
      key_len: 4
          ref: NULL
         rows: 15
     filtered: 100.00
        Extra: Using where; Using index
3 rows in set, 1 warning (0.00 sec)
```

※実行計画としてサブクエリのマテリアライズが選択されている。8.0.20 までは「DEPENDENT SUBQUERY」だった。

※データを約 10 万行まで増やして時間計測。いずれもデータがバッファプールに載っている状態で実行。

```
mysql> SET AUTOCOMMIT=0;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE t1 SET t1.key = 0 WHERE t1.key IN (SELECT /*+ SUBQUERY(INTOEXISTS) */ t2.key FRO
M t2 WHERE t2.val IN (40, 60, 100));
Query OK, 24576 rows affected (2.71 sec)
Rows matched: 24576  Changed: 24576  Warnings: 0
```

※マテリアライズを無効化して実行。実行計画は 8.0.20 以前と同じ。3 回実行した平均は 2.78 秒。

```
mysql> ROLLBACK;
Query OK, 0 rows affected (0.95 sec)

mysql> SET AUTOCOMMIT=0;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE t1 SET t1.key = 0 WHERE t1.key IN (SELECT t2.key FROM t2 WHERE t2.val IN (40, 60,
100));
Query OK, 24576 rows affected (1.08 sec)
Rows matched: 24576  Changed: 24576  Warnings: 0
```

※マテリアライズ有効（デフォルト）で実行。3 回実行した平均は 1.07 秒。2〜3 倍高速化した。

```
mysql> ROLLBACK;
Query OK, 0 rows affected (0.99 sec)
```

A.5.3　Redo ログの無効化が可能に

初期データロード（バックアップデータのリストアなど）の時間短縮が可能になりました。

【注】通常稼働時の性能向上を目的とした機能ではありません。

公式リファレンスマニュアル

- https://dev.mysql.com/doc/refman/8.0/en/innodb-redo-log.html#innodb-disable-redo-logging

ブログ記事等

- https://atsuizo.hatenadiary.jp/entry/2020/07/16/140000

A.5.4　Undo テーブルスペースの処理性能向上と安定化・ACID Undo DDL のサポート

公式リファレンスマニュアル

- https://dev.mysql.com/doc/refman/8.0/en/innodb-undo-tablespaces.html

A.5.5　テーブルスペースのパス検証の無効化が可能に

公式リファレンスマニュアル

- https://dev.mysql.com/doc/refman/8.0/en/innodb-disabling-tablespace-path-validation.html

A.6　その他の変更と新機能（第 10 章）

A.6.1　KEY パーティショニングでカラムインデックスプレフィックスを使ったときに正しく警告・エラーを出力するようになった

公式リファレンスマニュアル

- https://dev.mysql.com/doc/refman/8.0/en/partitioning-limitations.html#partitioning-limitations-prefixes

A.6.2　文字列型と数値型・時間型の値を比較する際に SQL 標準に準拠した型変換を行うようになった

公式リファレンスマニュアル

- https://dev.mysql.com/doc/relnotes/mysql/8.0/en/news-8-0-21.html
  - 「Building on work done in MySQL 8.0.18, …」