GMA022D0

GridDB® Advanced Edition
SQL リファレンス

東芝ソリューション株式会社

は じ め に

本書では、GridDB Advanced Edition における SQL の記述方法および、注意事項について記載しています。
GridDB Advanced Edition をご使用になる前に、必ずお読みください。
なお、本書で説明する機能は、GridDB Advanced Edition ライセンスを保有するユーザのみがご利用いただけます。

商標

目次

1. GridDB Advanced Edition とは

GridDB Advanced Edition では、GridDB のデータに SQL でアクセスできるインターフェース(NewSQL インタフェース)を提供します。

本書では、GridDB Advanced Edition（以降　GridDB AE と記載します）のサポートするデータベースにアクセスするための NewSQL インタフェースの SQL について説明します。NoSQL インタフェースとは異なるインタフェースですのでご注意ください。

なお、JDBC ドライバの使用方法は、「GridDB Advanced Edition JDBC ドライバ説明書」を参照してください。

2. SQL 記述形式

本章では、NewSQL インタフェースで使用できる SQL の記述形式について示します。

2.1 使用できる操作

本バージョンでは SELECT 文の他、CREATE TABLE 等の DDL(Data Definition Language、データ定義言語)や INSERT/DELETE 文などをサポートしています。詳細は第 3 章を参照して下さい。

2.2 データ型

2.2.1 データ格納に使用する型と NULL の格納

NewSQL インタフェースでデータの格納に使用する型を表 1 に示します。この型名はテーブル作成時にカラム型として記述できます。
NoSQL インタフェースのクライアントでは INTEGER 型は 32 ビットの整数値ですが、NewSQL インタフェースの INTEGER 型は 64 ビットの整数値であり、格納できる値の範囲が異なります。

表 1：データの格納に使用する型と NULL の扱い

データ型	内容詳細	NULL の扱い
BOOL 型	True/False	-
BYTE 型	-2^7 から 2^7-1 (8 ビット)の整数値	0
SHORT 型	-2^{15} から $2^{15}-1$ (16 ビット)の整数値	0
INTEGER 型	-2^{31} から $2^{31}-1$ (32 ビット)の整数値	0
LONG 型	-2^{63} から $2^{63}-1$ (64 ビット)の整数値	0
FLOAT 型	倍精度型(64 ビット) IEEE754 で定められた浮動小数点数	0.0
DOUBLE 型	倍精度型(64 ビット) IEEE754 で定められた浮動小数点数	0.0
TimeStamp 型	日付と時刻の組。 日付のみ、時刻のみの値を格納・取得する場合、時刻 00:00:00 または、日付 1970-01-01 が指定されたとみなす。	-
STRING 型	Unicode コードポイントを文字とする、任意個数の文字の列より構成されるテキスト文字	サイズ 0 のテキスト
BLOB 型	画像や音声などのバイナリデータのためのデータ型。入力したままの形式で保存されるラージオブジェクト。文字 x あるいは X をつけて、X'23AB'のような 16 進表現もできます。	サイズ 0 のデータ

また、本バージョンではテーブルに NULL を格納することができません。NULL を格納しようとした場合には、表 1 に示す値が格納されます。ただし SQL 式での演算では NULL を使用することができます。

これにより、“IS NULL”などの演算子をテーブルに格納された値に対して使用すると SQL 仕様とは異なる結果となる場合がありますのでご注意下さい。

2.2.2 テーブル作成時にカラム型として記述可能な表現

NewSQL インタフェースでは、テーブル作成時にカラム型として記述された型名について、2.2.1 で列挙した型名と一致しなくても、ルールに従って解釈しデータの格納に使用する型を決定します。

以下のルールを上から順にチェックし、合致したルールによってデータ格納に使用する型を決定します。ルールのチェック時には記述した型名およびルールでチェックする文字列の大文字小文字は区別しません。複数のルールに合致した場合はより上にあるルールが優先されます。どのルールにも当てはまらない場合はエラーとなりテーブル作成に失敗します。

ルール No.	テーブル作成時、カラム型として記述した文字列	作成するテーブルのカラム型
1	2.2.1 に列挙した型名	テーブル作成時に指定された型に従う
2	"REAL"	DOUBLE 型
3	"TINYINT"から始まる型名	BYTE 型
4	"SMALLINT"から始まる型名	SHORT 型
5	"BIGINT"から始まる型名	LONG 型
6	"INT"を含む型名	INTEGER 型
7	"CHAR", "CLOB", "TEXT"のどれかを含む型名	STRING 型
8	"BLOB"を含む型名	BLOB 型
9	"REAL", "DOUB"のどれかを含む型名	DOUBLE 型
10	"FLOA"を含む型名	FLOAT 型

上記ルールによるデータ型決定の例を示します。

- 記述した型名が"BIGINTEGER"→LONG 型(ルール 5)
- 記述した型名が"LONG"→LONG 型(ルール 1)
- 記述した型名が"TINYINT"→BYTE 型(ルール 3)
- 記述した型名が"FLOAT"→FLOAT 型(ルール 1)
- 記述した型名が"VARCHAR"→TEXT 型(ルール 7)
- 記述した型名が"CHARINT"→LONG 型(ルール 6)
- 記述した型名が"BIGBLOB"→BLOB 型(ルール 8)
- 記述した型名が"FLOATDOUB"→DOUBLE 型(ルール 9)
- 記述した型名が"INTREAL"→LONG 型(ルール 6)
- 記述した型名が"FLOATINGPOINT"→LONG 型(ルール 6)
- 記述した型名が"DECIMAL"→エラー

NoSQL インタフェースのクライアントにおけるデータ型と同等の型を NewSQL インタフェースで使用する場合は以下のように記述してください。ただし一部同等の型が存在せず、使用できないものがあります。

表 2：NoSQL インタフェースのクライアントのデータ型と同等のカラム型記述

NoSQL インタフェースのデータ型	同等の型となる NewSQL インタフェースのカラム型記述
STRING(文字列型)	"STRING" または "String 型となる表現"
BOOL(ブール型)	"BOOL"
BYTE(8 ビット整数型)	"BYTE" または "BYTE 型となる表現"
SHORT(16 ビット整数型)	"SHORT" または "SHOR 型となる表現"
INTEGER(32 ビット整数型)	INTEGER
LONG(64 ビット整数型)	"LONG" または "LONG 型となる表現"
FLOAT(32 ビット単精度浮動小数点数型)	"FLOAT" または "FLOAT 型となる表現"
DOUBLE(64 ビット倍精度浮動小数点数型)	"DOUBLE" または "DOUBLE 型となる表現"
TIMESTAMP(時刻型)	"TIMESTAMP"
GEOMETRY(空間型)	存在しません(使用できません)
BLOB 型	"BLOB" または "BLOB 型となる表現"
配列型	存在しません(使用できません)

2.2.3 コンテナをテーブルとしてアクセスするときのデータ型と値の扱い

NoSQL インタフェースのクライアントで作成したコンテナを、NewSQL インタフェースでテーブルとしてアクセスする場合のコンテナのカラム型および値の扱いを以下に示します。

表 3：コンテナのデータ型と値の NewSQL インタフェースでの扱い

コンテナのカラム型	NewSQL にマッピングされるデータ型	値
STRING 型	STRING 型	元の値と同一
BOOL 型	BOOL 型	元の値と同一
BYTE 型	BYTE 型	元の値と同一
SHORT 型	SHORT 型	元の値と同一
INTEGER 型	INTEGER 型	元の値と同一
LONG 型	LONG 型	元の値と同一
FLOAT 型	FLOAT 型	元の値と同一
DOUBLE 型	DOUBLE 型	元の値と同一
TIMESTAMP 型	TIMESTAMP 型	元の値と同一
GEOMETRY 型	STRING 型	全ての値が NULL
BLOB 型	BLOB 型	元の値と同一
配列型	STRING 型	全ての値が NULL

2.3 ユーザとデータベース

GridDB のユーザには、管理ユーザと一般ユーザの 2 種類があり、利用できる機能に違いがあります。また、データベースを作成することで、利用ユーザ単位にアクセスを分離することができます。ユーザ、データベースの詳細は「テクニカルリファレンス」をご参照ください。

さらに、NewSQL インタフェースでデータを登録、検索するには、データを格納するテーブル（表）を作成する必要があります。登録されるデータは行と呼び、1 個以上の列データから構成されます。

データベース名・列名および一般ユーザ名は、1 文字以上の ASCII 英数字ならびにアンダースコア「_」の列で構成されます。ただし、先頭の文字には数字を指定できません。

テーブル名も同様ですが、ノードアフィニティ機能向けに「@」の文字も指定できます。ノードアフィニティ機能については「テクニカルリファレンス」をご参照ください。

なお、テーブル名、列名は大文字・小文字の区別がありません。一方、データベース名と一般ユーザ名は命名時の大文字・小文字は保持されますが、大文字小文字同一視した場合に同一名となるデータベース、一般ユーザは作成できません。

3. GridDB AE でサポートされる SQL 文

サポートされる SQL 文は、表 4 の通りです。

表 4：サポートされる SQL 文一覧

コマンド	概要
CREATE DATABASE	データベースを作成する。
CREATE TABLE	テーブルを作成する。
CREATE INDEX	索引を作成する。
CREATE USER	一般ユーザを作成する。
DROP DATABASE	データベースを削除する。
DROP TABLE	テーブルを削除する。
DROP INDEX	索引を削除する。
DROP USER	一般ユーザを削除する。
SET PASSWORD	一般ユーザのパスワードを変更する。
GRANT	一般ユーザにデータベースへのアクセス権を設定する。
REVOKE	一般ユーザからデータベースへのアクセス権を削除する。
INSERT	テーブルに行を挿入する。
DELETE	テーブルから行を削除する。
UPDATE	テーブルにある行を更新する。
SELECT	データを取得する。
Comment	コメントを表記する。

本章では、SQL 文の分類ごとに説明を行います。

3.1 データ定義言語(DDL)

CREATE 文、DROP 文などから構成されます。

3.1.1 CREATE DATABASE

データベースを作成します。

形式
```
CREATE DATABASE データベース名;
```

- 管理ユーザのみが実行可能です。
- 「public」、「information_schema」と同一の名前のデータベースは、GridDB の内部用に予約済みのため作成できません。
- 既に同一の名前のデータベースが存在した場合は何も変更しません。
  データベース名に使用できる文字は、ASCII の英数字とアンダースコア(「_」)のみです。 ASCII の大文字と小文字は同一視されます。先頭には数字を使用できません。

3.1.2 CREATE TABLE

テーブルを作成します。

```
形式
  CREATE TABLE [IF NOT EXISTS] テーブル名 ( 列定義 1※ [, 列定義 2 …] )
    [PARTITION BY HASH 列名 PARTITIONS 分割数];

  ※列定義: 列名 型名 [列制約※]
  ※列制約: [PRIMARY KEY] (先頭の列のみ指定可)
```

- “IF NOT EXISTS”が指定された場合、指定したテーブル名と同じ名前のテーブルが存在しないときのみ作成します。
- “列定義”では、列名と型名の指定が必須です。
  - 指定可能な型名は 2.2 データ型を参照してください。
- “列制約”は、先頭の列の[PRIMARY KEY]のみサポートしています。
- “PARTITION BY HASH 列名 PARTITIONS 分割数”が指定された場合、指定した列の値によりデータパーティショニングされたテーブル（ラージテーブル）を作成します。
  - 「列名」に指定する列は、String 型か INTEGER 型である必要があります。
  - [PRIMARY KEY]を指定した列がある場合、「列名」はそれと同一である必要があります。
  - 「分割数」は、2 以上かつ 10000 以下である必要があります。
- テーブル名・列名に使用できる文字は、ASCII の英数字とアンダースコア(「_」)のみです。 ASCII の大文字と小文字は同一視されます。テーブル名・列名の先頭には数字を使用できません。
- テーブル名に「@ヒント情報」を付けることでノードアフィニティ機能を使うことができます。ノードアフィニティ機能については「テクニカルリファレンス」をご参照ください。
- 通常のテーブルは NoSQL の単一のコレクションとして作成され、データパーティショニングされたテーブル（ラージテーブル）は NoSQL の複数のコレクションとして作成されます。テーブルを NoSQL の時系列として作成することはできません。

3.1.3 CREATE INDEX

索引を作成します。

```
形式
  CREATE INDEX [IF NOT EXISTS] インデックス名 ON テーブル名 ( 索引をつける列名 );
```

- 既に存在する索引と同じ名前の索引は作成できません。
- 処理対象のテーブルにおいて実行中のトランザクションが存在する場合、それらの終了を待機してから作成を行います。
- インデックス名に使用できる文字は、ASCII の英数字とアンダースコア(「_」)のみです。 ASCII の大文字と小文字は同一視されます。先頭には数字を使用できません。

3.1.4 CREATE USER

一般ユーザを作成します。

```
形式
  CREATE USER ユーザ名 IDENTIFIED BY ‘パスワード文字列’;
```

- 管理ユーザのみが実行可能です。
- インストール時に登録済の管理ユーザ(admin および system)と同一の名前のユーザは作成できません。
- ユーザ名に使用できる文字は、ASCII の英数字とアンダースコア(「_」)のみです。 ASCII の大文字と小文字は同一視されます。先頭には数字を使用できません。
- パスワード文字列に使用できる文字は、ASCII 文字のみです。大文字と小文字は区別されます。
- “gs#”で始まるテーブル、”information_schema”と同名のテーブルは作成できません。

3.1.5 DROP DATABASE

データベースを削除します。

```
形式
  DROP DATABASE データベース名;
```

- 管理ユーザのみが実行可能です。
- 「public」、「information_schema」という名前のデータベース、及び「gs#」で始まる名前のデータベースは、GridDB の内部用に予約済みのため削除できません。
- ユーザが作成したテーブルが存在するデータベースは削除できません。

3.1.6 DROP TABLE

テーブルを削除します。

```
形式
  DROP TABLE テーブル名;
```

- 処理対象のテーブルにおいて実行中のトランザクションが存在する場合、それらの終了を待機してから削除を行います。

3.1.7 DROP INDEX

指定された索引を削除します。

```
形式
  DROP INDEX [IF EXISTS] インデックス名;
```

- “IF EXISTS”が指定された場合、指定した名前の索引が存在しない場合は何も変更しません。
- 処理対象のテーブルにおいて実行中のトランザクションが存在する場合、それらの終了を待機してから削除を行います。

3.1.8 DROP USER

一般ユーザを削除します。

```
形式
  DROP USER ユーザ名;
```

- 管理ユーザのみが実行可能です。

3.1.9 SET PASSWORD

一般ユーザのパスワードを変更します。

```
形式
  SET PASSWORD [FOR ユーザ名] = ‘パスワード文字列’;
```

- 管理ユーザは全ての一般ユーザのパスワードを変更可能です。
- 一般ユーザは自身のパスワードのみ変更可能です。

3.2 データ制御言語(DCL)

3.2.1 GRANT 文

一般ユーザにデータベースへのアクセス権を付与します。

```
形式
  GRANT ALL ON データベース名 TO ユーザ名;
```

- 管理ユーザのみが実行可能です。
- 1 つのデータベースにアクセス可能な一般ユーザは 1 ユーザに制限されています。

3.2.2 REVOKE 文

一般ユーザからデータベースへのアクセス権を剥奪します。

```
形式
  REVOKE ALL ON データベース名 FROM ユーザ名;
```

- 管理ユーザのみが実行可能です。

3.3 データ操作言語(DML)

3.3.1 INSERT 文

テーブルに行を登録します。

```
形式
  {INSERT|INSERT OR REPLACE|REPLACE} INTO テーブル名
    {VALUES ( {数値 1|文字列 1} [, {数値 2|文字列 2} …] ), … | SELECT 文};
```

3.3.2 DELETE 文

テーブルから行を削除します。

```
形式
  DELETE FROM テーブル名 [WHERE 抽出条件];
```

3.3.3 UPDATE 文

テーブルに存在する行を更新します。

```
形式
  UPDATE テーブル名
    SET 列名 1 = 式 1 [, 列名 2 = 式 2 …]
    [WHERE 抽出条件];
```

- パーティショニングを設定した場合、UPDATE を使ってパーティションキーになっている項目を別の値に更新することは出来ません。このような場合は、DELETE 後に INSERT を行ってください。

  (例)

  create table tab(a integer, b string) partition by hash a partitions 5;

  NG : update tab set a = a * 2;

  [240015:SQL_COMPILE_PARTITIONING_KEY_NOT_UPDATABLE] Partitioning column='a' is not datable on executing statement

  OK: update tab set b = 'XXX';

3.3.4 SELECT 文

データを取得します。

```
形式
  SELECT [{ALL|DISTINCT}]  列名 1 [, 列名 2 …] [FROM 句]
    [WHERE 句]
    [GROUP BY 句 [HAVING 句]]
    [ORDER BY 句]
    [LIMIT 句 [OFFSET 句]];
```

FROM 句、WHERE 句など様々な句から構成されます。

3.4 句

3.4.1 FROM 句

SELECT を行うテーブル名を指定します。

```
形式
  FROM テーブル名 1 [, テーブル名 2 … ]
```

3.4.2 GROUP BY 句

前に指定された句の結果の中で、指定された列で同じ値を持った行をグループ化します。

```
形式
  GROUP BY 列名 1 [,  列名 2 …]
```

3.4.3 HAVING 句

GROUP BY 句によりグループ化された情報に対して探索条件で絞り込みを行います。GROUP BY 句は省略できません。

```
形式
  HAVING 探索条件
```

3.4.4 ORDER BY 句

検索結果の並べ替え（ソート）を行います。

```
形式
  ORDER BY 列名 1 [{ASC|DESC}]  [, 列名 2 [{ASC|DESC}] …]
```

3.4.5 WHERE 句

先行する FROM 句の結果に、探索条件を適用します。

```
形式
  WHERE 探索条件
```

- 探索条件

  探索条件は述語や SELECT 文などが使用できます。

3.4.6 LIMIT 句/OFFSET 句

指定した位置から指定した件数分のデータを取り出します。

```
形式
  LIMIT 値 1 OFFSET 値 2
```

値 1 は取り出すデータ件数を表し、値 2 は取り出すデータ位置を表します。

3.5 述語

比較演算子(=、>など)を使った比較述語以外に BETWEEN 述語、IN 述語、LIKE 述語を使うことができます。

3.5.1 BETWEEN 述語

指定した範囲の値を取り出します。

```
形式
  式 1 [NOT] BETWEEN 式 2 AND 式 3
```

BETWEEN 述語が真となるのは、次の条件のときです。

式 2 ≦ 式 1 ≦ 式 3

NOT を指定した場合は、条件を満足しない行に対して、この述語は真になります。

3.5.2 IN 述語

条件を満たす集合を取り出します。

```
形式
  式 1 [NOT] IN ( 式 2 [, 式 3 …] )
```

3.5.3 LIKE 述語

パターン一致比較を行います。

```
形式
  式 [NOT] LIKE 文字パターン [ESCAPE エスケープ文字]
```

文字パターンは、%や_の特殊文字を使って表現します。

%：任意の文字列

_：任意の文字

なお、%や_を通常の文字として使いたい場合には、エスケープ文字を「ESCAPE エスケープ文字」形式で指定した上で、エスケープ文字を%や_の前に記述してください。

3.6 Comment

SQL 文中にコメントを書くことができます。
書式は、--（ハイフンを 2 つ）の後ろに記述するか、/* */で囲みます。コメントの後ろには改行が必要です。

3.7 関数

GridDB AE の SQL 文には以下の関数が用意されています。

AVG、GROUP_CONCAT、SUM、TOTAL、EXISTS、ABS、CHAR、COALESCE、IFNULL、INSTR、HEX、LENGTH、
LIKE、LOWER、LTRIM、MAX、MIN、NULLIF、PRINTF、QUOTE、RANDOM、RANDOMBLOB、REPLACE、
ROUND、RTRIM、SUBSTR、TRIM、TYPEOF、UNICODE、UPPER、ZEROBLOB、NOW、TIMESTAMP、
TO_TIMESTAMP_MS、TO_EPOCH_MS、EXTRACT、TIMESTAMPADD、TIMESTAMPDIFF

A　参考文献

- 日本工業標準調査会ウェブサイト, http://www.jisc.go.jp/, JISX3005-2 データベース言語ＳＱＬ第２部：基本機能（ＳＱＬ／Ｆｏｕｎｄａｔｉｏｎ）

B　付録：予約語

GridDB AE の SQL では、以下の単語が予約語として定義されています。

ABORT ACTION AFTER ALL ANALYZE AND AS ASC BEGIN BETWEEN BY CASE CAST COLLATE
COLUMN COMMIT CONFLICT CREATE CROSS DATABASE DAY DELETE DESC DISTINCT DROP ELSE

END ESCAPE EXCEPT EXCLUSIVE EXISTS EXPLAIN EXTRACT FALSE FOR FROM GLOB GRANT
GROUP HASH HAVING HOUR IDENTIFIED IF IN INDEX INITIALLY INNER INSERT INSTEAD
INTERSECT INTO IS ISNULL JOIN KEY LEFT LIKE LIMIT MATCH MILLISECOND MINUTE
MONTH NATURAL NO NOT NOTNULL NULL OF OFFSET ON OR ORDER OUTER PARTITION
PARTITIONS PASSWORD PLAN PRAGMA PRIMARY QUERY RAISE REGEXP RELEASE REPLACE
RESTRICT REVOKE RIGHT ROLLBACK ROW SECOND SELECT SET TABLE THEN TIMESTAMPADD
TIMESTAMPDIFF TO TRANSACTION TRUE UNION UPDATE USER USING VALUES VIEW VIRTUAL
WHEN WHERE WITHOUT XOR YEAR