GaussDB

Oracle Compatibility(Centralized)

Issue	01
Date	2025-04-14

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Oracle Database Compatibility Overview

GaussDB is generally compatible with Oracle Database in terms of basic functions (such as data types, SQL statements, and database objects) and PL/SQL. However, due to architecture design differences, there are still some incompatible items.

This chapter compares the Oracle-compatible mode in GaussDB 505.2.1 with Oracle Database 19c.

2 Basic SQL Elements

2.1 Data Types

Table 2-1 Numeric types

No.	Oracle Database	GaussDB	Difference
1	NUMBER [(p [, s])]	Supported, with differences.	The precision and usage are different. • When NUMBER contains parameters, the maximum boundary values of precision *p* and scale *s* in GaussDB are greater than those in Oracle Database. • In GaussDB, the default value of *p* when NUMBER does not contain parameters is much greater than the maximum boundary value when NUMBER contains parameters. However, in Oracle Database, the former is equal to the latter. • In GaussDB, the value of *s* cannot be negative. In Oracle Database, a negative *s* value is accurate to an integer.
2	FLOAT [(p)]	Supported.	-
3	BINARY_FLOAT	Not supported.	-
4	BINARY_DOUBLE	Supported.	-

Table 2-2 Date and time types

No.	Oracle Database	GaussDB	Difference
1	DATE	Supported, with differences.	The precision is different. GaussDB supports a larger time range than Oracle Database.
2	TIMESTAMP [(fractional_seconds_precision)]	Supported, with differences.	-
3	TIMESTAMP [(fractional_seconds_precision)] WITH TIME ZONE	Supported, with differences.	The timestamptz type of GaussDB is equivalent to the timestampwithlocaltimezone type of Oracle Database. The type corresponding to timestamptz of Oracle Database is missing. Time zone update: In some countries or regions, the time zone information is often updated. Therefore, the database system often needs to modify the time zone file accordingly to ensure that the time is correct. Currently, the GaussDB time zone type involves only timestamp with timezone. When a new time zone file takes effect, the existing data is not changed, and the new data is adjusted based on the time zone file information. The capability of the same type of data in the database is different from that in the Oracle Database.
4	TIMESTAMP [(fractional_seconds_precision)] WITH LOCAL TIME ZONE	Not supported.	-
5	INTERVAL YEAR [(year_precision)] TO MONTH	Supported.	-

No.	Oracle Database	GaussDB	Difference
6	INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)]	Supported.	-

NOTE

- In A-compatible mode, the DATE type is replaced by TIMESTAMP(0) WITHOUT TIME ZONE. The differences between DATE and TIMESTAMP(0) WITHOUT TIME ZONE are the same.
- In terms of TIMESTAMP [(fractional_seconds_precision)] and TIMESTAMP [(fractional_seconds_precision)] WITH TIME ZONE, the differences between GaussDB and Oracle Database are as follows:
  - The value of **fractional_seconds_precision** ranges from 0 to 6 in GaussDB, but ranges from 0 to 9 in Oracle Database.
  - GaussDB uses **DateStyle** to set the display format of date and time values and the rules of resolving ambiguous values. For details, see "SQL Reference > Data Type > Date/Time Types" in *Developer Guide*. Generally, the input format verification and output display in Oracle Database are controlled by the **NLS_TIMESTAMP_FORMAT** and **NLS_TIMESTAMP_TZ_FORMAT** parameters.
  - By default, GaussDB removes zeros from the end of the decimal part of the second. Oracle Database controls the display of the decimal part based on the setting (**FF/FF1-FF9**) of the formatting parameter. For example, '2017-09-01 10:32:19.212000' is displayed as '2017-09-01 10:32:19.212' in GaussDB. In Oracle Database, it is displayed as '2017-09-01 10:32:19.212' if the **format** parameter contains FF, or '2017-09-01 10:32:19.212000000' if the **format** parameter contains **FF9**.
  - GaussDB supports a wider time range than Oracle Database.

Table 2-3 Character types

No.	Oracle Database	GaussDB	Difference
1	VARCHAR2 (size [BYTE | CHAR])	Supported, with differences.	In GaussDB, the unit of size is byte. That is, only **BYTE** is supported. You cannot select a value between **BYTE** and **CHAR**. The maximum size is 10 MB. In Oracle Database, however, the unit of size can be selected between **BYTE** and **CHAR**. If **MAX_STRING_SIZE** is set to **EXTENDED**, the maximum size is 32767 bytes. If **MAX_STRING_SIZE** is set to **STANDARD**, the maximum size is 4000 bytes. The actual number of characters that can be contained depends on the character set in use.

No.	Oracle Database	GaussDB	Difference
2	NVARCHAR2 (size)	Supported, with differences.	In GaussDB, NVARCHAR2(*n*) is the alias of VARCHAR2(*n*). In Oracle Database, NVARCHAR2(*n*) is different from VARCHAR2(*n*)**. When MAX_STRING_SIZE** is set to **EXTENDED**, the maximum size is 32767 bytes. When **MAX_STRING_SIZE** is set to **STANDARD**, the maximum size is 4000 bytes. The actual number of characters that can be contained depends on the character set in use.
3	CHAR [(size [BYTE | CHAR])]	Supported, with differences.	In GaussDB, the unit of size is byte. That is, only **BYTE** is supported. You cannot select a value between **BYTE** and **CHAR**. The maximum size is 10 MB. In Oracle Database, however, the unit of size can be selected between **BYTE** and **CHAR**. The maximum size is 2000 bytes. The actual number of characters that can be contained depends on the character set in use.
4	NCHAR [(size)]	Supported, with differences.	In GaussDB, the unit of size is byte, and the maximum size is 10 MB. In Oracle Database, however, the unit of size is character, and the maximum size is 2000 bytes. The actual number of characters that can be contained depends on the character set in use.
5	CLOB	Supported, with differences.	Locators are not supported.
6	NCLOB	Not supported.	-
7	LONG	Not supported.	-

Table 2-4 Binary types

No.	Oracle Database	GaussDB	Difference
1	RAW (size)	Supported, with differences.	In GaussDB, **size** indicates the recommended byte length and is not used to verify the byte length of the input raw type.
2	LONG RAW	Not supported.	-
3	BLOB	Supported.	-
4	BFILE	Not supported.	-

Table 2-5 ROWID types

No.	Oracle Database	GaussDB
1	ROWID	Not supported.
2	UROWID	Not supported.

Table 2-6 User-defined types

No.	Oracle Database	GaussDB
1	Object types	Not supported.
2	REF data types	Not supported.
3	Variable arrays	Supported.
4	Nested tables	Supported.

Table 2-7 Pseudo-types

No.	Oracle Database	GaussDB
1	anytype	Not supported.
2	anydata	Not supported.
3	anydataset	Not supported.

Table 2-8 XML types

No.	Oracle Database	GaussDB	Difference
1	XMLTYPE	Supported, with differences.	GaussDB does not support some operations. For example, the XMLELEMENT function is used to convert a character string to the XML type instead of the XMLType type. For details, see "SQL Reference > Data Type > XMLTYPE" in *Developer Guide*.
2	URIType	Not supported.	-

Table 2-9 Spatial types

No.	Oracle Database	GaussDB
1	SDO_GEOMETRY	Not supported.
2	SDO_TOPO_GEOMETRY	Not supported.
3	SDO_GEORASTER	Not supported.

Table 2-10 Lock modes

No.	Oracle Database	GaussDB
1	none	-
2	null	AccessShare
3	RS	RowShare
4	RX	RowExclusive
5	S	ShareUpdateExclusive
6	SRX	Share
7	-	ShareRowExclusive
8	X	Exclusive
9	-	AccessExclusive
10	-	INVALID **NOTE** INVALID of GaussDB indicates that an invalid lock is assigned. An invalid lock is assigned only when a lock that cannot be identified by GaussDB occurs during system running.

2.2 Data Type Comparison Rules

Data type comparison (collation) rules are followed when values of the same data type are compared (collated).

Table 2-11 Comparison rules

No.	Oracle Database	GaussDB	Difference
1	Numeric values	Supported.	-
2	Datetime values	Supported.	-
3	Binary values	Supported.	-
4	Character values	Supported, with differences.	• GaussDB and Oracle Database support different comparison rules, and the names of the same comparison rules may be different. • GaussDB and Oracle Database differ in specifying comparison rules. For example, table-level comparison rules cannot be specified in GaussDB, but can be specified in Oracle Database. • GaussDB and Oracle Database differ in the syntax for specifying comparison rules. For example, in GaussDB, the **ENCODING**, **LC_CTYPE**, and **LC_COLLATE** parameters are used to specify the character set, character type, and comparison rules used during database creation. For details, see "SQL Reference > SQL Syntax > C > CREATE DATABASE" in *Developer Guide*. In Oracle Database, comparison rules at different levels are usually specified by a series of parameters with the NLS prefix.
5	Object values	Not supported.	-
6	Varrays and nested tables	Supported, with differences.	Both Oracle Database and GaussDB support the comparison of varrays. Different from Oracle Database, GaussDB not only supports the comparison of the number of elements in two varrays, but also supports the comparison between varrays of the same type.

No.	Oracle Database	GaussDB	Difference
7	Data type precedence	Supported.	-
8	Explicit/ Implicit data conversion	Supported.	-

2.3 Literals

Table 2-12 Literals

No.	Oracle Database	GaussDB
1	Text literals	Supported.
2	Numeric literals	Supported.
3	Datetime literals	Supported.
4	Interval literals	Supported.

2.4 Format Models

Table 2-13 Formats

No.	Oracle Database	GaussDB	Difference
1	Number formats	Supported, with differences.	GaussDB supports the $, C, TM, TM9, TME, and U formats only when **a_format_version** is set to **10c** and **a_format_dev_version** is set to **s1**. In addition, this parameter does not support the TH, PL, or SG format. For details about GaussDB, see the **fmt parameter of the number type** table in "SQL Reference > Functions and Operators > Type Conversion Functions" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
2	Datetime formats	Supported, with differences.	GaussDB: Parameters used for time truncation and rounding are valid only when **a_format_version** is set to **10c** and **a_format_dev_version** is set to **s1**. For details about GaussDB support, see formats for formatting date and time in "SQL References > Functions and Operators > Date and Time Processing Functions and Operators" in *Developer Guide*.
3	Format model modifiers	Supported.	-
4	String-to-date conversion rules	Supported, with differences.	GaussDB: The to_timestamp_tz function is valid only when **a_format_version** is set to **10c** and **a_format_dev_version** is set to **s1**. For details about GaussDB, see to_date, to_timestamp, and to_timestamp_tz in "SQL Reference > Functions and Operators > Type Conversion Functions" in *Developer Guide*.
5	XML format models	Not supported.	-

2.5 Nulls

Table 2-14 Nulls

No.	Oracle Database	GaussDB
1	IS NULL and IS NOT NULL	Supported.
2	NULLS in conditions	Supported.

2.6 Comments

Table 2-15 Comments

No.	Oracle Database	GaussDB	Difference
1	A slash and an asterisk (/*)	Supported.	-
2	Two hyphens (--)	Supported.	-
3	**COMMENT** command	Supported.	-
4	HINT	Supported, with differences.	GaussDB does not support the '--+' hint format. For details, see "SQL Optimization > Hint-based Tuning" in *Developer Guide*.

2.7 Database Objects

Table 2-16 Schema objects

No.	Oracle Database	GaussDB	Difference
1	Analytic views	Not supported.	-
2	Attribute dimensions	Not supported.	-
3	Clusters	Supported.	-
4	Constraints	Supported.	-
5	Database links	Supported.	-
6	Database triggers	Supported.	-
7	Dimensions	Supported.	-

No.	Oracle Database	GaussDB	Difference
8	External procedure libraries	Not supported.	-
9	Hierarchies	Not supported.	-
10	Index-organized tables	Not supported.	-
11	Indexes	Supported.	-
12	Index types	Not supported.	-
13	Java classes	Not supported.	-
14	Java resources	Not supported.	-
15	Java source code	Not supported.	-
16	Join groups	Not supported.	-
17	Materialized views	Supported.	-
18	Materialized view logs	Not supported.	-
19	Mining models	Not supported.	-
20	Object tables	Not supported.	-
21	Object types	Not supported.	-
22	Object views	Not supported.	-
23	Operators	Supported.	-
24	Packages	Supported.	-
25	Sequences	Supported.	-

No.	Oracle Database	GaussDB	Difference
26	Storage functions	Supported.	-
27	Stored procedures	Supported.	-
28	Synonyms	Supported, with differences.	The names of Oracle Database objects in the same namespace must be unique. In GaussDB, the name of a synonym can be the same as that of a table, view, function, or package in the same namespace. In this case, GaussDB preferentially accesses the table, view, function, or package object with the same name. If no such object is found, GaussDB searches for the synonym object. In addition, the PUBLIC synonym is searched only when the schema name of the object to which the synonym points is a username. For details about the search sequence, see "SQL Reference > SQL Syntax > C > CREATE SYNONYM" in *Developer Guide*.
29	Tables	Supported.	-
30	Views	Supported.	-
31	Zone map	Not supported.	-

Table 2-17 Non-schema objects

No.	Oracle Database	GaussDB
1	Contexts	Not supported.
2	Directories	Supported.
3	Editions	Not supported.
4	Flashback archives	Not supported.
5	Lockdown profiles	Not supported.
6	Profiles	Not supported.
7	Restore points	Supported.
8	Roles	Supported.

No.	Oracle Database	GaussDB
9	Rollback segments	• Ustore supports rollback segments. • Astore does not support rollback segments.
10	Tablespaces	Supported.
11	Tablespace sets	Not supported.
12	Unified audit policies	Supported.
13	Users	Supported.

2.8 Database Object Names and Qualifiers

Table 2-18 Naming rules

No.	Oracle Database	GaussDB	Difference
1	Database object naming rules	Supported, with differences.	GaussDB uses lowercase letters by default.
2	Schema object naming rules	Supported.	-

2.9 Syntax for Schema Objects and Parts in SQL Statements

Table 2-19 Object reference

No.	Oracle Database	GaussDB
1	General syntax for referring to an object	Supported.
2	Resolving a reference to an object	Supported.
3	Referring to objects in other schemas	Supported.
4	Referring to objects in remote databases	Supported.
5	Referring to partitions and subpartitions of tables and indexes	Supported.

3 Pseudocolumns

GaussDB is compatible with hierarchical query pseudocolumns, sequence pseudocolumns, and rownum pseudocolumns. Other pseudocolumns are not supported.

Hierarchical Query Pseudocolumns

Table 3-1 Hierarchical query pseudocolumns

No.	Oracle Database	GaussDB
1	connect_by_iscycle	Supported.
2	connect_by_isleaf	Supported.
3	LEVEL pseudocolumn	Supported.

Sequence Pseudocolumns

Table 3-2 Sequences

No.	Oracle Database	GaussDB	Difference
1	currval	Supported, with differences.	It is implemented as a function in GaussDB. The call mode is compatible with Oracle Database.
2	nextval	Supported, with differences.	It is implemented as a function in GaussDB. The call mode is compatible with Oracle Database.

ROWNUM Pseudocolumn

Table 3-3 rownum

No.	Oracle Database	GaussDB	Difference
1	rownum	Supported, with differences.	When Oracle Database uses rownum in the left, right, and full join conditions for filtering, the performance varies according to the conditions. The rownum condition may be ignored or partially ignored. However, GaussDB filters the results after left, right, and full join.

4 Operators

GaussDB is basically compatible with operators in Oracle Database.

SQL Operators

Table 4-1 SQL operators

No.	Oracle Database	GaussDB
1	Unary and binary operators	Supported.
2	Operator precedence	Supported.

Arithmetic Operators

Table 4-2 Arithmetic operators

No.	Oracle Database	GaussDB
1	Unary operators: positive (+) and negative (–).	Supported.
2	Binary operators: addition (+) and subtraction (–).	Supported.
3	Binary operators: multiplication (*) and division (/).	Supported.

COLLATE Operator

Table 4-3 COLLATE operator

No.	Oracle Database	GaussDB
1	COLLATE collation_name	Supported.

Connection Operators

Table 4-4 Connection operators

No.	Oracle Database	GaussDB
1	||	Supported.

Hierarchical Query Operators

Table 4-5 Hierarchical query operators

No.	Oracle Database	GaussDB	Difference
1	prior	Supported, with differences.	GaussDB: Only common columns can be called. Functions cannot be called.
2	connect_by_root	Supported, with differences.	GaussDB: When connect_by_root is called, if parentheses are used to modify the operation value, the behavior is the same as that of Oracle Database. If parentheses are not used, this operator can be called only for common columns.
3	connect_by_iscycle	Supported, with differences.	It can be used independently in GaussDB and must be used together with NOCYCLE in Oracle Database.
4	start with, connect by, and level	Supported, with differences.	In GaussDB, start with, connect by, and level can be used as column aliases, but they cannot be used as column aliases in Oracle Database.

No.	Oracle Database	GaussDB	Difference
5	start with	Supported, with differences.	In GaussDB, start with can be followed by an empty string for query, and the query result is empty. Operations on pseudocolumns are supported. In Oracle Database, an error is reported in these cases.
6	connect by	Supported, with differences.	In GaussDB, connect by can be used with NULL values. In Oracle Database, an error is reported.

Set Operators

Table 4-6 Set operators

No.	Oracle Database	GaussDB
1	union	Supported.
2	union all	Supported.
3	intersect	Supported.
4	minus	Supported.

Multiset Operators

Table 4-7 Multiset operators

No.	Oracle Database	GaussDB
1	multiset except	Supported.
2	multiset intersect	Supported.
3	multiset union	Supported.

User-defined Operators

Table 4-8 User-defined operators

No.	Oracle Database	GaussDB	Difference
1	CREATE OPERATOR	Supported, with differences.	• Oracle Database provides CONTEXT_CLAUSE to define functional evaluation functions, which is different from GaussDB that constrains selectivity evaluation functions. GaussDB does not support user-defined functional evaluation functions. • Optional parameters in Oracle Database differ greatly from those in GaussDB. For details, see the GaussDB parameter description in "SQL Reference > SQL Syntax > C > CREATE OPERATOR" in *Developer Guide*.

Comparison Operators

No.	Oracle Database	GaussDB
1	< =	Supported.
2	< >	Supported.
3	> =	Supported.
4	^ =	Supported.
5	! =	Not supported. For !=, if there is a space between an exclamation mark (!) and an equal sign (=), the exclamation mark will be identified as factorial.

For comparison operators <=, <>, >=, and ^=, if there is a space between two symbols, it does not affect normal operations. For !=, if there is a space between an exclamation mark (!) and an equal sign (=), the exclamation mark will be identified as factorial, which may cause the result to be inconsistent with the expected result.

5 Expressions

GaussDB is compatible with most expressions in Oracle Database.

Table 5-1 Expressions

No.	Oracle Database	GaussDB	Difference
1	Simple expressions	Supported.	-
2	Analytic view expressions	Not supported.	-
3	Compound expressions	Supported.	-
4	CASE expressions	Supported.	-
5	Column expressions	Supported.	-
6	CURSOR expressions	Not supported.	-
7	Datetime expressions	Supported, with differences.	GaussDB command output does not contain time zone information, but Oracle Database contains time zone information similar to "PM AMERICA/LOS_ANGELES."
8	Function expressions	Supported.	-

No.	Oracle Database	GaussDB	Difference
9	Interval expressions	Partially supported.	GaussDB supports statements in the format of "SELECT INTERVAL '999999999 23:59:59.999' day(9) to second FROM DUAL;" but does not support statements in the format of "SELECT(SYSDATE-SYSDATE) DAY TO SECOND FROM DUAL;". They are supported in Oracle Database.
10	JSON object access expressions	Partially supported, with differences.	1. GaussDB can extract values from JSON objects in "->'key'" mode, while Oracle Database can extract values in ".key" mode. 2. For JSONARRY objects, Oracle Database can extract values corresponding to all keys at a time in ".key" mode. However, GaussDB does not support this function.
11	Model expressions	Not supported.	-
12	Object expressions	Not supported.	-
13	Placeholder expressions	Partially supported.	GaussDB supports general placeholder expressions such as ":var", but does not support the combination of two general placeholder expressions using the INDICATOR keyword.
14	Scalar subquery expressions	Supported.	-
15	Type constructor expressions	Partially supported.	GaussDB cannot specify the NEW keyword before the type constructor, but Oracle Database can.
16	Expression lists	Supported.	-

6 Conditions

This chapter describes common compatible conditions. The conditions include comparison, floating-point, logical, model, multi-set, pattern matching, NULL value, XML, SQL/JSON, composite, BETWEEN, EXISTS, IN, and IS OF TYPE. For details, see **Table 6-1**.

Table 6-1 Conditions

No.	Oracle Database	GaussDB	Difference
1	Comparison conditions	Supported, with differences.	Differences exist when statements contain the ANY, SOME, and ALL operators. Oracle Database supports operations on list objects, but GaussDB needs to convert list objects into array expressions before performing operations.
2	Floating-point conditions	Not supported.	-
3	Logical conditions	Supported.	-
4	Model conditions	Not supported.	-
5	Multiset conditions	Not supported.	-
6	Pattern-matching conditions	Supported.	-
7	NULL conditions	Supported.	-
8	XML conditions	Not supported.	-

No.	Oracle Database	GaussDB	Difference
9	SQL/JSON conditions	Partially supported, with differences.	• GaussDB does not support the IS JSON and JSON_TEXTCONTAINS conditions. • The JSONB_EQ condition in GaussDB is the same as the JSON_EQUAL condition in Oracle Database. However, GaussDB does not support the ERROR clause. • The JSONB_EXISTS condition in GaussDB is the same as the JSON_EXISTS condition in Oracle Database. However, GaussDB does not support the ERROR, EMPTY, or PASSING clauses.
10	Compound conditions	Supported.	-
11	BETWEEN condition	Supported.	-
12	EXISTS condition	Supported.	-
13	IN condition	Supported.	-
14	IS OF TYPE condition	Not supported.	-

7 Drivers

7.1 JDBC

7.1.1 Array

This section describes the differences between Oracle Database and GaussDB when the JDBC driver of the java.sql.Array type is used.

Table 7-1 Constructor reference

Constructor	Oracle Database	GaussDB	API Difference
1. Use the static constructor of ArrayDescriptor to construct an ArrayDescriptor object. 2. Use ArrayDescriptor to construct an array object.	String typeName = "XXX"; Connection conn = getConnection(); Object[] elements = null; ArrayDescriptor desc = ArrayDescriptor.createDescriptor(typeName, conn); Array array = new ARRAY(desc, conn, elements);	String typeName = "xxx"; Connection conn = getConnection(); Object[] elements = null; ArrayDescriptor desc = ArrayDescriptor.getDescriptor(typeName, conn); Array array = new GaussArray(desc, elements);	• Different names of the static constructor for ArrayDescriptor: In Oracle Database, it is createDescriptor, while in GaussDB it is getDescriptor. • Different constructor names for arrays: In Oracle Database, it is ARRAY(ArrayDescriptor, Connection, Object), while in GaussDB, it is GaussArray(ArrayDescriptor, Object). • Variable description: **typeName** indicates the type name and is case-sensitive. Generally, it uses uppercase in Oracle Database, while in GaussDB it is lowercase. **conn** indicates the connection object for the corresponding database. **elements** indicates the element data of the corresponding type.

NOTE

1. The constructors that are not listed in the preceding table are not supported by GaussDB.
2. If the element type is a character type and the length of the construction input string exceeds that defined by the element type, Oracle Database reports an error during input parameter binding.

   If the array type is the varray type and the number of elements exceeds the maximum length of varray, Oracle Database reports an error during input parameter binding.

   GaussDB does not verify type modifiers when constructing or binding input parameters. When a database receives array objects and executes SQL statements, it decides whether to report an error.

Table 7-2 API reference

Method	Return Value Type	Throws	GaussDB
getBaseTypeName()	String	SQLException	Supported.
getBaseType()	int	SQLException	Supported.
getArray()	Object	SQLException	Supported.
getArray(java.util.Map<String,Class<?>> map)	Object	SQLException	Not supported.
getArray(long index, int count)	Object	SQLException	Supported.
getArray(long index, int count, java.util.Map<String,Class<?>> map)	Object	SQLException	Not supported.
getResultSet()	ResultSet	SQLException	Not supported.
getResultSet(java.util.Map<String,Class<?>> map)	ResultSet	SQLException	Not supported.
getResultSet(long index, int count)	ResultSet	SQLException	Not supported.
getResultSet (long index, int count, java.util.Map<String,Class<?>> map)	ResultSet	SQLException	Not supported.
free()	void	SQLException	Not supported.

Table 7-3 Differences in the getArray() API

Element Database Type	Actual Return Value Type of the getArray API (Oracle Database)	Actual Return Value Type of the getArray API (GaussDB)
CHAR	java.lang.String[]	java.lang.String[]
VARCHAR/ VARCHAR2	java.lang.String[]	java.lang.String[]
NCHAR	java.lang.String[]	java.lang.String[]
NVARCHAR2	java.lang.String[]	java.lang.String[]
NUMBER	java.math.BigDecimal[]	java.math.BigDecimal[]
NUMERIC	java.math.BigDecimal[]	java.math.BigDecimal[]
DECIMAL	java.math.BigDecimal[]	java.math.BigDecimal[]
INTEGER	java.math.BigDecimal[]	java.lang.Integer[]
SMALLINT	java.math.BigDecimal[]	java.lang.Short[]
DOUBLE PRECISION	java.math.BigDecimal[]	java.lang.Double[]
FLOAT	java.math.BigDecimal[]	java.lang.Double[]
REAL	java.math.BigDecimal[]	java.lang.Float[]
BINARY_DOUBLE	java.lang.Double[]	java.lang.Double[]
BINARY_INTEGER	java.math.BigDecimal[]	java.lang.Integer[]
BOOLEAN	java.math.BigDecimal[]	java.lang.Boolean[]
TIMESTAMP	java.sql.Timestamp[]	java.sql.Timestamp[]
TIMESTAMP WITH TIME ZONE	java.time.OffsetDateTime[]	java.sql.Timestamp[]
BLOB	oracle.sql.BLOB[]	java.sql.Blob[]
CLOB	oracle.sql.CLOB[]	java.sql.Clob[]
Set/Array	java.lang.Object[]	java.sql.Array[]
RECORD	java.lang.Object[]	java.sql.Struct[]

NOTE

1. GaussDB currently does not support the types unlisted in the aforementioned table.
2. For details about the differences in the return values of the getArray(long index, int count) API, see the preceding table.
3. The differences in **index** parameter of getArray(long index, int count) are as follows:
   - The value range supported by Oracle Database is [1, Long.MAX_VALUE]. GaussDB supports a range of [1, Integer.MAX_VALUE].
   - If the **index** value is greater than that of **Integer.MAX_VALUE**, it will be truncated in Oracle Database, while an error is reported in GaussDB.

Table 7-4 Differences in the getBaseType() API

Element Database Type	Return Value of the getBaseType API (Oracle Database)	Return Value of the getBaseType API (GaussDB)
CHAR	java.sql.Types.CHAR	java.sql.Types.CHAR
VARCHAR/ VARCHAR2	java.sql.Types.VARCHAR	java.sql.Types.VARCHAR
NCHAR	java.sql.Types.NCHAR	java.sql.Types.CHAR
NVARCHAR2	java.sql.Types.NVARCHAR	java.sql.Types.VARCHAR
NUMBER	java.sql.Types.NUMERIC	java.sql.Types.NUMERIC
NUMERIC	java.sql.Types.DECIMAL	java.sql.Types.NUMERIC
DECIMAL	java.sql.Types.DECIMAL	java.sql.Types.NUMERIC
INTEGER	java.sql.Types.NUMERIC	java.sql.Types.INTEGER
SMALLINT	java.sql.Types.NUMERIC	java.sql.Types.SMALLINT
DOUBLE PRECISION	java.sql.Types.FLOAT	java.sql.Types.DOUBLE
FLOAT	java.sql.Types.FLOAT	java.sql.Types.DOUBLE
REAL	java.sql.Types.FLOAT	java.sql.Types.REAL
BINARY_DOUBLE	oracle.jdbc.OracleTypes.BINARY_DOUBLE	java.sql.Types.DOUBLE
BINARY_INTEGER	java.sql.Types.NUMERIC	java.sql.Types.INTEGER
BOOLEAN	java.sql.Types.NUMERIC	java.sql.Types.BIT
TIMESTAMP	java.sql.Types.TIMESTAMP	java.sql.Types.TIMESTAMP
TIMESTAMP WITH TIME ZONE	oracle.jdbc.OracleTypes.TIMESTAMPTZ	java.sql.Types.TIMESTAMP
BLOB	java.sql.Types.BLOB	java.sql.Types.BLOB

Element Database Type	Return Value of the getBaseType API (Oracle Database)	Return Value of the getBaseType API (GaussDB)
CLOB	java.sql.Types.CLOB	java.sql.Types.CLOB
Set/Array	java.sql.Types.ARRAY	java.sql.Types.ARRAY
RECORD	java.sql.Types.STRUCT	java.sql.Types.STRUCT

NOTE

GaussDB currently does not support the types unlisted in the aforementioned table.

Table 7-5 Differences in the getBaseTypeName() API

Element Database Type	Return Value of the getBaseTypeName API (Oracle Database)	Return Value of the getBaseTypeName API (GaussDB)
CHAR	"CHAR"	"bpchar"
VARCHAR/ VARCHAR2	"VARCHAR"	"varchar"
NCHAR	"NCHAR"	"bpchar"
NVARCHAR2	"NVARCHAR"	"nvarchar2"
NUMBER	"NUMBER"	"numeric"
NUMERIC	"DECIMAL"	"numeric"
DECIMAL	"DECIMAL"	"numeric"
INTEGER	"NUMBER"	"int4"
SMALLINT	"NUMBER"	"int2"
DOUBLE PRECISION	"FLOAT"	"float8"
FLOAT	"FLOAT"	"float8"
REAL	"FLOAT"	"float4"
BINARY_DOUBLE	"BINARY_DOUBLE"	"float8"
BINARY_INTEGER	"NUMBER"	"int4"
BOOLEAN	"NUMBER"	"bool"
TIMESTAMP	"TIMESTAMP"	"timestamp"
TIMESTAMP WITH TIME ZONE	"TIMESTAMP WITH TIME ZONE"	"timestamptz"

Element Database Type	Return Value of the getBaseTypeName API (Oracle Database)	Return Value of the getBaseTypeName API (GaussDB)
BLOB	"BLOB"	"blob"
CLOB	"CLOB"	"clob"
Set/Array	See the description below.	See the description below.
RECORD	See the description below.	See the description below.

NOTE

1. GaussDB currently does not support the types unlisted in the aforementioned table.
2. When an element is of the set, array, or RECORD type that is defined within a package, the return rules for getBaseTypeName are as follows:
   - OJDBC11 returns *schemaName.packageName.typeName*.
   - OJDBC8 generally returns *schemaName.packageName.typeName*, or returns *"schemaName"."packageName.typeName"* in the following condition:

     Any of *schemaName*, *packageName*, or *typeName* does not meet the rule of starting with a letter followed by characters including letters, digits, or underscores.
   - GaussDB generally returns *schemaName.packageName.typeName*, or returns *"schemaName"."packageName"."typeName"* in the following condition:

     Any of *schemaName*, *packageName*, or *typeName* does not meet the rule of starting with a letter or underscore followed by characters including letters, digits, or underscores.
3. When an element is of the set, array, or RECORD type that is defined in the schema (but not in the package), the return rules for getBaseTypeName are as follows:
   - OJDBC11 returns *schemaName.typeName*.
   - OJDBC8 generally returns *schemaName.typeName*, or returns *"schemaName"."typeName"* in the following condition:

     Any of *schemaName* or *typeName* does not meet the rule of starting with a letter followed by characters including letters, digits, or underscores.
   - GaussDB generally returns *schemaName.typeName*, or returns *"schemaName"."typeName"* in the following condition:

     Any of *schemaName* or *typeName* does not meet the rule of starting with a letter or underscore followed by characters including letters, digits, or underscores.
4. If no special processing is performed during element type creation, the getBaseTypeName API typically returns the type name in uppercase in Oracle Database, while returns the type name in lowercase in GaussDB.

Table 7-6 Differences in array construction APIs

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
CHAR	Any Java type	Byte, Short, Integer, Long, BigInteger, BigDecimal, Float, Double, Character, Boolean, java.sql.Date, java.sql.Time, java.sql.Timestamp, and PGClob	The different types supported by element input parameters can be seen in the table.
VARCHAR/ VARCHAR2	Any Java type	Byte, Short, Integer, Long, BigInteger, BigDecimal, Float, Double, Character, Boolean, java.sql.Date, java.sql.Time, java.sql.Timestamp, and PGClob	The different types supported by element input parameters can be seen in the table.
NCHAR	Any Java type	Byte, Short, Integer, Long, BigInteger, BigDecimal, Float, Double, Character, Boolean, java.sql.Date, java.sql.Time, java.sql.Timestamp, and PGClob	The different types supported by element input parameters can be seen in the table.
NVARCHAR2	Any Java type	Byte, Short, Integer, Long, BigInteger, BigDecimal, Float, Double, Character, Boolean, java.sql.Date, java.sql.Time, java.sql.Timestamp, and PGClob	The different types supported by element input parameters can be seen in the table.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. When the input parameter is of the Float, Double, BigDecimal, or String type and the decimal part is 0, Oracle Database truncates the decimal part, while GaussDB retains it.
NUMERIC	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. When the input parameter is of the Float, Double, BigDecimal, or String type and the decimal part is 0, Oracle Database truncates the decimal part, while GaussDB retains it.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
DECIMAL	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. When the input parameter is of the Float, Double, BigDecimal, or String type and the decimal part is 0, Oracle Database truncates the decimal part, while GaussDB retains it.
INTEGER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getArray() API. 3. When the input parameter value exceeds the integer range, GaussDB reports an error.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
SMALLINT	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getArray() API. 3. When the input parameter exceeds the Short range, GaussDB reports an error.
DOUBLE PRECISION	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getArray() API. 3. Converting a higher-precision type to Double may result in precision loss.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
FLOAT	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getArray() API. 3. Converting a higher-precision type to Double may result in precision loss.
REAL	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getArray() API. 3. Converting a higher-precision type to Float may result in precision loss.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
BINARY_DOUBLE	Double and oracle.sql.BINARY_DOUBLE	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getArray() API. 3. Converting a higher-precision type to Double may result in precision loss.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
BINARY_INTEGER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, AtomicInteger, AutomicLong, DoubleAccumulator, DoubleAddr, LongAccumulator, LondAdder, Striped64, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. The input parameters of the OJDBC11 element support the following types: Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER. 3. The target types are inconsistent. For details, see differences in the getArray() API. 4. When the input value exceeds the Integer range, GaussDB reports an error. Oracle Database OJDBC8 performs data truncation.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
BOOLEAN	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, AtomicInteger, AutomicLong, DoubleAccumulator, DoubleAddr, LongAccumulator, LondAdder, Striped64, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by element input parameters can be seen in the table. 2. The input parameters of the OJDBC8 element support the following types: Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER. 3. The target types are inconsistent. For details, see differences in the getArray() API. 4. In GaussDB, the target data type is Boolean, which only supports inputs of **1**, **0**, **"true"**, and **"false"**.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
TIMESTAMP	byte[], java.sql.Date, Calendar, java.util.Date, LocalDate, LocalDateTime, LocalTime, OffsetDateTime, OffsetTime, String, java.sql.Time, java.sql.Timestamp, oracle.sql.DATE, oracle.sql.TIMESTAMP, oracle.sql.TIMESTAMPTZ, oracle.sql.TIMESTAMPLTZ, and ZonedDateTime	java.util.Date, java.sql.Date, java.sql.Time, java.sql.Timestamp, LocalDateTime, and String	The different types supported by element input parameters can be seen in the table.
TIMESTAMP WITH TIME ZONE	java.sql.Date, Calendar, java.util.Date, LocalDate, LocalDateTime, LocalTime, OffsetDateTime, OffsetTime, String, java.sql.Time, java.sql.Timestamp, oracle.sql.DATE, oracle.sql.TIMESTAMP, oracle.sql.TIMESTAMPTZ, oracle.sql.TIMESTAMPLTZ, and ZonedDateTime	java.util.Date, java.sql.Date, java.sql.Time, java.sql.Timestamp, LocalDateTime, and String	1. The different types supported by element input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getArray() API.
BLOB	oracle.sql.BLOB and oracle.jdbc.driver.OracleBlob	PGBlob	The different types supported by element input parameters can be seen in the table.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
CLOB	oracle.sql.CLOB, oracle.jdbc.driver.OracleClob, InputStream, and Reader	PGClob	The different types supported by element input parameters can be seen in the table.
Set/Array	Array and Object	GaussArray and Object	1. The different types supported by element input parameters can be seen in the table. 2. Oracle Database: No error is reported when the input parameter is of the Array type even if the Array type differs from the element type. GaussDB: An error is reported when the input parameter is of the GaussArray type that differs from the element type. 3. When the input parameter of an element is of the Object type, refer to the differences in array construction APIs.

Element Database Type	List of Java Types Supported by Element Input Parameters (Oracle Database OJDBC8)	List of Java Types Supported by Element Input Parameters (GaussDB)	Difference
RECORD	Struct and Object[]	GaussStruct and Object[]	1. The different types supported by element input parameters can be seen in the table. 2. Oracle Database: No error is reported when the input parameter is of the Struct type even if the Struct type differs from the element type. GaussDB: An error is reported when the input parameter is of the GaussStruct type that differs from the element type. 3. When the input parameter of an element is of the Object[] type, refer to the differences in struct construction APIs.

NOTE

1. When an array is constructed, if the Java type of the input element does not match the target type, an implicit conversion operation is performed. For details on the Java types of input elements supported by various database element types, refer to the preceding table.
2. GaussDB currently does not support the types unlisted in the aforementioned table.
3. The constructor needs to provide an element array. The preceding table describes the differences between elements in the array.

7.1.2 Struct

This section describes the differences between Oracle Database and GaussDB when the JDBC driver of the java.sql.Struct type is used.

Table 7-7 Constructor reference

Constructor	Oracle Database	GaussDB	API Difference
1. Use the static constructor of StructDescriptor to construct a StructDescriptor object. 2. Use StructDescriptor to construct a struct object.	String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; StructDescriptor desc = StructDescriptor.createDescriptor(typeName, conn); Struct struct = new STRUCT(desc, conn, attributes);	String typeName = "xxx"; Connection conn = getConnection(); Object[] elements = null; StructDescriptor desc = StructDescriptor.getDescriptor(typeName, conn); Struct struct = new GaussStruct(desc, attributes);	• Different names of the static constructor for StructDescriptor: In Oracle Database, it is createDescriptor, while in GaussDB it is getDescriptor. • Different constructor names for structs. In Oracle Database, it is defined as STRUCT(StructDescriptor, Connection, Object[]), while in GaussDB, it is defined as GaussStruct(StructDescriptor, Object[]). • Variable description: **typeName** indicates the type name and is case-sensitive. Generally, it uses uppercase in Oracle Database, while in GaussDB it is lowercase. **conn** indicates the connection object for the corresponding database. **attributes** indicates the element data array.

Constructor	Oracle Database	GaussDB	API Difference
Use the createStruct standard API of Connection to construct a struct object.	String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; Struct struct = conn.createStruct(typeName, attributes);	String typeName = "XXX"; Connection conn = getConnection(); Object[] attributes = null; Struct struct = conn.createStruct(typeName, attributes);	• Variable description: **typeName** indicates the type name and is case-sensitive. Generally, it uses uppercase in Oracle Database, while in GaussDB it is lowercase. **conn** indicates the connection object for the corresponding database. **attributes** indicates the element data array.

NOTE

1. GaussDB currently does not support the constructors not listed in the aforementioned table.
2. If the attribute type is a character type and the length of the construction input string exceeds that defined by the element type, Oracle Database reports an error during input parameter binding.

   GaussDB does not verify type modifiers when constructing or binding input parameters. When a database receives struct objects and executes SQL statements, it decides whether to report an error.
3. If the number of array elements exceeds the actual number of columns of the corresponding type, an error is reported during creation.

   When the number of array elements is less than the actual number of columns, the creation of Oracle Database is successful, but an error is reported during parameter input for execution; GaussDB reports an error during creation.

Table 7-8 API reference

Method	Return Value Type	Throws	GaussDB
getSQLTypeName()	String	SQLException	Supported.
getAttributes()	Object[]	SQLException	Supported.

Method	Return Value Type	Throws	GaussDB
getAttributes(java.util.Map<String,Class<?>> map)	Object[]	SQLException	Not supported.

NOTE

The differences in the getSQLTypeName API are as follows:

1. For the package type, the struct constructed in the *packageName.typeName* format, the differences in the getSQLTypeName API are as follows:
   - OJDBC11 returns *packageName.typeName*.
   - OJDBC8 generally returns *packageName.typeName*, or returns *"packageName"."typeName"* when *packageName* and *typeName* meet the following condition:

     Any of *packageName* or *typeName* does not meet the rule of starting with a letter followed by characters including letters, digits, or underscores.
   - GaussDB generally returns *schemaName.packageName.typeName*, or returns *"schemaName"."packageName"."typeName"* when *schemaName*, *packageName*, and *typeName* meet the following condition:

     Any of *schemaName*, *packageName*, or *typeName* does not meet the rule of starting with a letter or underscore followed by characters including letters, digits, or underscores.
2. For the package type, the differences in the getSQLTypeName API in other scenarios are as follows:
   - OJDBC11 returns *schemaName.packageName.typeName*.
   - OJDBC8 generally returns *schemaName.packageName.typeName*, or returns *"schemaName"."packageName.typeName"* when *schemaName*, *packageName*, and *typeName* meet the following condition:

     Any of *schemaName*, *packageName*, or *typeName* does not meet the rule of starting with a letter followed by characters including letters, digits, or underscores.
   - GaussDB generally returns *schemaName.packageName.typeName*, or returns *"schemaName"."packageName"."typeName"* when *schemaName*, *packageName*, and *typeName* meet the following condition:

     Any of *schemaName*, *packageName*, or *typeName* does not meet the rule of starting with a letter or underscore followed by characters including letters, digits, or underscores.
3. For the non-package type, the differences in the getSQLTypeName API are as follows:
   - OJDBC11 returns *schemaName.typeName*.
   - OJDBC8 generally returns *schemaName.typeName*, or returns *"schemaName"."typeName"* when *schemaName* and *typeName* meet the following condition:

     Any of *schemaName* or *typeName* does not meet the rule of starting with a letter followed by characters including letters, digits, or underscores.
   - GaussDB generally returns *schemaName.typeName*, or returns *"schemaName"."typeName"* when *schemaName* and *typeName* meet the following condition:

     Any of *schemaName* or *typeName* does not meet the rule of starting with a letter or underscore followed by characters including letters, digits, or underscores.

Table 7-9 Differences in the getAttributes() API

Database Attribute Type	Java Type of the Corresponding Element in the Return Value (Oracle Database OJDBC8)	Java Type of the Corresponding Element in the Return Value (Oracle Database OJDBC11)	Java Type of the Corresponding Element in the Return Value (GaussDB)
CHAR	String	String	String
VARCHAR/ VARCHAR2	String	String	String
NCHAR	String	String	String
NVARCHAR2	String	String	String
NUMBER	BigDecimal	BigDecimal	BigDecimal
NUMERIC	BigDecimal	BigDecimal	BigDecimal
DECIMAL	BigDecimal	BigDecimal	BigDecimal
INTEGER	BigDecimal	BigDecimal	Integer
SMALLINT	BigDecimal	BigDecimal	Short
DOUBLE PRECISION	BigDecimal	BigDecimal	Double
FLOAT	BigDecimal	BigDecimal	Double
REAL	BigDecimal	BigDecimal	Float
BINARY_DOUBLE	Double	Double	Double
BINARY_INTEGER	BigDecimal	Integer	Integer
BOOLEAN	BigDecimal	Integer	Boolean
TIMESTAMP	Timestamp	Timestamp	Timestamp
TIMESTAMP WITH TIME ZONE	TIMESTAMPTZ	TIMESTAMPTZ	Timestamp
BLOB	BLOB	BLOB	PGBlob
CLOB	CLOB	CLOB	PGClob
Set/Array	ARRAY	ARRAY	GaussArray
RECORD	STRUCT	STRUCT	GaussStruct

NOTE

GaussDB currently does not support the types unlisted in the aforementioned table.

Table 7-10 Differences in struct construction APIs

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
CHAR	Any Java type	Byte, Short, Integer, Long, BigInteger, BigDecimal, Float, Double, Character, Boolean, String, java.sql.Date, java.sql.Time, java.sql.Timestamp, and PGClob	1. The different types supported by attribute input parameters can be seen in the table. 2. When the input parameter is of the String type, Oracle Database adds spaces at the end of the string until its length matches the length defined by the type; GaussDB does not add spaces.
VARCHAR/ VARCHAR2	Any Java type	Byte, Short, Integer, Long, BigInteger, BigDecimal, Float, Double, Character, Boolean, String, java.sql.Date, java.sql.Time, java.sql.Timestamp, and PGClob	The different types supported by attribute input parameters can be seen in the table.
NCHAR	Any Java type	Byte, Short, Integer, Long, BigInteger, BigDecimal, Float, Double, Character, Boolean, String, java.sql.Date, java.sql.Time, java.sql.Timestamp, and PGClob	The different types supported by attribute input parameters can be seen in the table.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
NVARCHAR2	Any Java type	Byte, Short, Integer, Long, BigInteger, BigDecimal, Float, Double, Character, Boolean, String, java.sql.Date, java.sql.Time, java.sql.Timestamp, and PGClob	The different types supported by attribute input parameters can be seen in the table.
NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. When the input parameter is of the Float, Double, BigDecimal, or String type and the decimal part is 0, Oracle Database truncates the decimal part, while GaussDB retains it.
NUMERIC	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. When the input parameter is of the Float, Double, BigDecimal, or String type and the decimal part is 0, Oracle Database truncates the decimal part, while GaussDB retains it.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
DECIMAL	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. When the input parameter is of the Float, Double, BigDecimal, or String type and the decimal part is 0, Oracle Database truncates the decimal part, while GaussDB retains it.
INTEGER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API. 3. When the input parameter value exceeds the integer range, GaussDB reports an error.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
SMALLINT	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API. 3. When the input parameter exceeds the Short range, GaussDB reports an error.
DOUBLE PRECISION	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API. 3. Converting a higher-precision type to Double may result in precision loss.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
FLOAT	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API. 3. Converting a higher-precision type to Double may result in precision loss.
REAL	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API. 3. Converting a higher-precision type to Float may result in precision loss.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
BINARY_DOUBLE	byte[], Double, and oracle.sql.BINARY_DOUBLE	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API. 3. Converting a higher-precision type to Double may result in precision loss.
BINARY_INTEGER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, AtomicInteger, AutomicLong, DoubleAccumulator, DoubleAddr, LongAccumulator, LondAdder, Striped64, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. If the input value exceeds the integer range, GaussDB reports an error, whereas Oracle Database truncates the value.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
BOOLEAN	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, AtomicInteger, AutomicLong, DoubleAccumulator, DoubleAddr, LongAccumulator, LondAdder, Striped64, String, and oracle.sql.NUMBER	Byte, Short, Integer, Long, Float, Double, Boolean, BigDecimal, BigInteger, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API. 3. In GaussDB, the target data type is Boolean, which only supports inputs of **1**, **0**, **"true"**, and **"false"**.
TIMESTAMP	byte[], java.sql.Date, String, java.sql.Time, java.sql.Timestamp, oracle.sql.TIMESTAMP, and oracle.sql.DATE	java.util.Date, java.sql.Date, java.sql.Time, java.sql.Timestamp, LocalDateTime, and String	The different types supported by attribute input parameters can be seen in the table.
TIMESTAMP WITH TIME ZONE	java.sql.Date, Calendar, java.util.Date, LocalDate, LocalDateTime, LocalTime, OffsetDateTime, OffsetTime, String, java.sql.Time, java.sql.Timestamp, oracle.sql.TIMESTAMP, oracle.sql.TIMESTAMPTZ, oracle.sql.TIMESTAMPLTZ, and ZonedDateTime	java.util.Date, java.sql.Date, java.sql.Time, java.sql.Timestamp, LocalDateTime, and String	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
BLOB	oracle.sql.BLOB and oracle.jdbc.driver.OracleBlob	PGBlob	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API.
CLOB	oracle.sql.CLOB and oracle.jdbc.driver.OracleClob	PGClob	1. The different types supported by attribute input parameters can be seen in the table. 2. The target types are inconsistent. For details, see differences in the getAttributes() API.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
Set/Array	Array and Object	GaussArray and Object	1. The different types supported by attribute input parameters can be seen in the table. 2. In Oracle Database, no error is reported when the attribute input parameter is of the Array type, even if the Array type differs from the actual type required by the attribute. In GaussDB, an error is reported when the attribute input parameter is of the GaussArray type that differs from the actual type required by the attribute. 3. When the input parameter of an attribute is of the Object type, refer to the differences in array construction APIs.

Database Attribute Type	List of Java Types Supported by Attribute Input Parameters (Oracle Database)	List of Java Types Supported by Attribute Input Parameters (GaussDB)	Difference
RECORD	Struct and Object[]	GaussStruct and Object[]	1. The different types supported by attribute input parameters can be seen in the table. 2. In Oracle Database, no error is reported when the attribute input parameter is of the Struct type, even if the Struct type differs from the actual type required by the attribute. In GaussDB, an error is reported when the attribute input parameter is of the GaussStruct type that differs from the actual type required by the attribute. 3. When the input parameter of an attribute is of the Object[] type, refer to the differences in struct construction APIs.

NOTE

1. When a struct is constructed, if the Java type of the input element does not match the target type, an implicit conversion operation is performed. For details on the Java types of input elements supported by various database element types, refer to the preceding table.
2. GaussDB currently does not support the types unlisted in the aforementioned table.
3. The constructor needs to provide an attribute array. The preceding table describes the differences of each attribute in the array.

8 Common SQL DDL Clauses

This chapter describes common compatible SQL DDL clauses, including allocate_extent_clause, constraint, deallocate_unused_clause, file_specification, logging_clause, parallel_clause, physical_attributes_clause, size_clause, storage_clause, and aggregate function nesting. For details, see **Table 8-1**.

Table 8-1 Common SQL DDL clauses

No.	Oracle Database	GaussDB	Difference
1	allocate_extent_clause Syntax: ALLOCATE EXTENT [({ SIZE size_clause | DATAFILE 'filename' | INSTANCE integer } ...)] For example, after the **employees** table is created, change the allocated extent **size** of the table to **10M**. SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), SALARY NUMBER(38), LAST_NAME VARCHAR2(16)); Table created. SQL> ALTER TABLE employees ALLOCATE EXTENT (SIZE 10M); Table altered.	Not supported.	-

No.	Oracle Database	GaussDB	Difference
2	constraint Syntax: { inline_constraint | out_of_line_constraint | inline_ref_constraint | out_of_line_ref_constraint } For example, when you create the **staff** table, the **ID** and **NAME** columns specified in the constraint clause cannot be empty. SQL> CREATE TABLE staff(ID INT NOT NULL, NAME char(8) NOT NULL, AGE INT, ADDRESS CHAR(50), SALARY REAL); Table created.	Supported.	-
3	deallocate_unused_clause Syntax: DEALLOCATE UNUSED [KEEP size_clause] For example, after creating the **employees** table and performing some INSERT and DELETE operations, you want to use the deallocate_unused_clause to release the unused space of the **employees** table. SQL> CREATE TABLE employees(EMPLOYEE_ID NUMBER(38), JOB_ID NUMBER(38), SALARY NUMBER(38), LAST_NAME VARCHAR2(16)); Table created. - Perform some INSERT and DELETE operations. SQL> ALTER TABLE employees DEALLOCATE UNUSED; Table altered.	Not supported.	-

No.	Oracle Database	GaussDB	Difference
4	file_specification Syntax: {['filename' | 'ASM_filename'] [SIZE size_clause] [REUSE] [autoextend_clause]} | {['filename | ASM_filename' | ('filename | ASM_filename' [, 'filename | ASM_filename']...)] [SIZE size_clause] [BLOCKSIZE size_clause [REUSE]} For example, to create a temporary tablespace **tbs_temp_01**, the file_specification clause of the SQL statement specifies that a temporary database file **templ01.dbf** is created in the tablespace. The tablespace can be automatically expanded and allocated to the tablespace group **tbs_grp_01**. SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01; Tablespace created.	Not supported.	-

No.	Oracle Database	GaussDB	Difference

5	logging_clause
	Syntax:
	{ LOGGING | NOLOGGING | FILESYSTEM_LIKE_LOGGING }
		Partially supported, with differences.	• GaussDB does not support the LOGGING and FILESYSTEM_LIKE_LOGGING constraint clauses.
			Example:
			When a table is created in GaussDB with the LOGGING constraint clause, a syntax error is reported.
			gaussdb=# CREATE LOGGING TABLE my_tab(id int, name char(16));
			ERROR: syntax error at or near "LOGGING"
			LINE 1: CREATE LOGGING TABLE my_tab(id int, name char(16));
			^
			When a table is created in GaussDB with the FILESYSTEM_LIKE_LOGGING constraint clause, a syntax error is reported.
			gaussdb=# CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name char(16));
			ERROR: syntax error at or near "FILESYSTEM_LIKE_LOGGING"
			LINE 1: CREATE FILESYSTEM_LIKE_LOGGING TABLE my_tab(id int, name cha...
			^
			• GaussDB supports only table-level UNLOGGED constraints and does not support column-level UNLOGGED constraints.
			For example, when a table is created in GaussDB with the column-level UNLOGGED constraint clause, a syntax error is reported.
			gaussdb=# CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16));
			ERROR: syntax error at or near "UNLOGGED"
			LINE 1: CREATE UNLOGGED TABLE my_tab(id int UNLOGGED, name char(16))...
			^
			• GaussDB uses logging clauses only in the CREATE TABLE, CREATE TABLE AS, and SELECT INTO statements.

No.	Oracle Database	GaussDB	Difference
			For example, when a TABLESPACE statement with the UNLOGGED constraint clause is created in GaussDB, a syntax error is reported. gaussdb=# CREATE UNLOGGED TABLESPACE tbs1 RELATIVE LOCATION 'tablespace1/tablespace_1'; ERROR: syntax error at or near "TABLESPACE" LINE 1: CREATE UNLOGGED TABLESPACE tbs1 RELATIVE LOCATION 'tablespac... ^
6	parallel_clause Syntax: { NOPARALLEL | PARALLEL [integer] } For example, if you create table **t1** and specify **PARALLEL 4** in the parallel_clause, a maximum of four parallel processes can be used to query and update table **t1**. SQL> CREATE TABLE t1 (id NUMBER, name VARCHAR2(50)) PARALLEL 4; Table created.	Not supported.	-

No.	Oracle Database	GaussDB	Difference

7	physical_attributes_clause
	Syntax:
	[{ PCTFREE integer | PCTUSED integer | INITRANS integer | storage_clause }...]
		Partially supported, with differences.	• GaussDB does not support PCTUSED.
			For example, if you run an SQL statement to create the **tbl1_ind** index in the **tbl1** table and set the space usage **PCTUSED** of the index to **20%** in the physical_attributes_clause of the statement, an error is reported when the SQL statement is executed in GaussDB.
			gaussdb=# CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20;
			ERROR: syntax error at or near "PCTUSED"
			LINE 1: CREATE INDEX tbl1_ind ON tbl1 (name) PCTUSED 20;
			^
			• GaussDB uses physical_attributes_clause only in the CREATE TABLE and CREATE INDEX statements.
			For example, if you run an SQL statement to obtain data from the **tbl1** table, create the materialized view **tbl1_mv**, and set the number of initial transactions of the view to **30** in the physical_attributes_clause, an error is reported when GaussDB executes the statement.
			gaussdb=# CREATE MATERIALIZED VIEW tbl1_mv INITRANS 30 as select * from tbl1;
			ERROR: syntax error at or near "INITRANS"
			LINE 1: CREATE MATERIALIZED VIEW tbl1_mv INITRANS 30 as select * fro...
			^

No.	Oracle Database	GaussDB	Difference
8	size_clause Syntax: integer [K | M | G | T | P | E] For example, create a temporary tablespace **tbs_temp_01** and a temporary database file **templ01.dbf** in the tablespace. The initial size of the tablespace is **5M** as specified by the size_clause in the SQL statement, which can be automatically expanded. The tablespace can be allocated to the tablespace group **tbs_grp_01**. SQL> CREATE TEMPORARY TABLESPACE tbs_temp_01 TEMPFILE 'temp01.dbf' SIZE 5M AUTOEXTEND ON TABLESPACE GROUP tbs_grp_01; Tablespace created.	Not supported.	-

No.	Oracle Database	GaussDB	Difference
9	storage_clause Syntax: STORAGE ({ INITIAL size_clause | NEXT size_clause | MINEXTENTS integer | MAXEXTENTS { integer | UNLIMITED } | maxsize_clause | PCTINCREASE integer | FREELISTS integer | FREELIST GROUPS integer | OPTIMAL [size_clause | NULL] | BUFFER_POOL { KEEP | RECYCLE | DEFAULT } | FLASH_CACHE { KEEP | NONE | DEFAULT } | (CELL_FLASH_CACHE (KEEP | NONE | DEFAULT)) | ENCRYPT } ...)	Partially supported, with differences.	• In Oracle Database, storage parameters are specified by the STORAGE clause. In GaussDB, storage parameters are specified by the WITH clause. Example: To create the **my_tab1** table in the Oracle Database, set the initial size of the table to **10M** in the storage_clause, and add 5 MB each time when more space is required, run the following SQL statement: SQL> CREATE TABLE my_tab1 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) STORAGE (INITIAL 10M NEXT 5M); Table created. To create the **my_tab2** table in GaussDB and set the storage engine type to **ustore** in the storage_clause, run the following SQL statement: gaussdb=# CREATE TABLE my_tab2 (id NUMBER(10) PRIMARY KEY, name VARCHAR2(50)) with (storage_type=ustore); NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "my_tab2_pkey" for table "my_tab2" CREATE TABLE • Optional storage parameters in GaussDB are greatly different from those in Oracle Database. For details, see the GaussDB parameter description in "SQL Reference > SQL Syntax > C > CREATE TABLE" in *Developer Guide*. **WITH ({storage_parameter = value} [, ...])** describes the storage parameters supported by the CREATE TABLE statement.

No.	Oracle Database	GaussDB	Difference
10	Aggregate function nesting For example, create the **revenue** table generated by nesting the aggregate functions MIN() and SUM() in the **sales_amount** column of the **sales** table. SQL> CREATE TABLE sales(ID INT, SALES_AMOUNT INT); Table created. SQL> INSERT INTO sales VALUES(1, 100); 1 row created. SQL> INSERT INTO sales VALUES (3, 200); 1 row created. SQL> CREATE TABLE revenue as SELECT SUM(MIN(sales_amount)) as total from sales group by sales_amount; Table created.	Supported.	-
11	Dropping a system schema Syntax: DROP USER schema_name CASCADE; For example, drop the SYS schema as the **SYS** user. SQL> DROP USER SYS; DROP USER SYS * ERROR at line 1: ORA-28050: specified user or role cannot be dropped	Supported.	-

9 SQL Queries and Subqueries

GaussDB is compatible with SQL queries and subqueries except hierarchical queries.

Table 9-1 SQL queries and subqueries

No.	Oracle Database	GaussDB	Difference
1	Creating simple queries	Supported.	-
2	Hierarchical queries	Supported, with differences.	GaussDB supports only Oracle Database's CONNECT_BY_FILTERING and does not support CONNECT_BY_NOFILTERING.
3	UNION [ALL], INTERSECT, MINUS	Supported.	-
4	Sorting query results	Supported, with differences.	If the GaussDB query does not contain groups and the target column contains both an aggregate function and a set returning function, the sorting of the set returning function column is not ignored.
5	Joins	Supported, with differences.	GaussDB supports only join types with Oracle Database, such as left/right, self, natural, and full outer join. Join optimization methods such as IN-MEMORY JOIN GROUP are not supported.
6	Using subqueries	Supported.	-

No.	Oracle Database	GaussDB	Difference
7	Unnesting of nested subqueries	Supported, with differences.	HASH_AJ or MERGE_AJ cannot be explicitly specified in GaussDB.
8	Distributed queries	Supported, with differences.	GaussDB needs to explicitly specify a database link for query.
9	Aggregate function nesting	Supported.	-

10 PL/SQL Language

GaussDB is compatible with PL/SQL operators, expressions, control statements, collections, and records, but does not support predefined PL/SQL constants, types, and subtypes.

10.1 Basic PL/SQL Syntax

Table 10-1 PL/SQL operators

No.	Oracle Database	GaussDB
1	+	Supported.
2	:=	Supported.
3	=>	Supported.
4	%	Supported.
5	'	Supported.
6	.	Supported.
7	||	Supported.
8	/	Supported.
9	**	Not supported.
10	(	Supported.
11	)	Supported.
12	:	Supported.
13	,	Supported.
14	<<	Supported.
15	>>	Supported.

No.	Oracle Database	GaussDB
16	/*	Supported.
17	*/	Supported.
18	*	Supported.
19	''	Supported.
20	..	Supported.
21	=	Supported.
22	<>	Supported.
23	!=	Supported.
24	~=	Supported.
25	^=	Supported.
26	<	Supported.
27	>	Supported.
28	<=	Supported.
29	>=	Supported.
30	@	Supported.
31	--	Supported.
32	;	Supported.
33	-	Supported.

Table 10-2 Logical operators

No.	Oracle Database	GaussDB
1	NOT	Supported.
2	AND	Supported.
3	OR	Supported.

Table 10-3 Comparison expressions

No.	Oracle Database	GaussDB
1	IS [NOT] NULL	Supported.
2	LIKE	Supported.

No.	Oracle Database	GaussDB
3	BETWEEN	Supported.
4	IN	Supported.

Table 10-4 CASE expressions

No.	Oracle Database	GaussDB
1	simple CASE	Supported.
2	searched CASE	Supported.

Table 10-5 Parameters related to variable declaration

No.	Oracle Database	GaussDB	Difference
1	%TYPE	Supported, with differences.	• GaussDB does not support *record%type*. • GaussDB does not support **pkg.***record%type* and **schema.pkg.***record%type* as the input and output parameter types. • In GaussDB, *Table/View*.**column.***column%type* or **schema.***Table/View*.**column.***column%type* cannot be nested with one or more layers as variable types or input/output parameter type. • In GaussDB, *record*.**column.***column%type* and **pkg.***record*.**column.***column%TYPE* cannot be nested with a column type of records at one or more layers as the variable type or input/output parameter type.

No.	Oracle Database	GaussDB	Difference
2	%ROWTYPE	Supported, with differences.	• When GaussDB has multiple CNs, the **%ROWTYPE** and **%TYPE** attributes of the temporary table cannot be declared in a stored procedure. The temporary table is valid only in the current session. During compilation, other CNs cannot view the temporary table of the current CN. Therefore, if there are multiple CNs, the system displays a message indicating that the temporary table does not exist. • GaussDB does not support *view%rowtype* and **schema.***view%rowtype* as the input and output parameter types. • GaussDB does not support the **package.***cursor%rowtype* as the input and output parameter types.

10.2 Data Type Compatibility

Table 10-6 Other PL/SQL data types

No.	Oracle Database	GaussDB	Difference
1	CHARACTER	Supported, with differences.	• GaussDB: The length ranges from 1 to 10485760 bytes. • Oracle Database: The length ranges from 1 to 32767 bytes.
2	VARCHAR	Supported, with differences.	• GaussDB: The length ranges from 1 to 10485760 bytes. • Oracle Database: The length ranges from 1 to 32767 bytes.
3	STRING	Not supported.	-
4	PLS_INTEGER	Not supported.	In GaussDB, you can use the INT type.
5	BINARY_INTEGER	Supported.	-

Table 10-7 User-defined PL/SQL subtypes

No.	Oracle Database	GaussDB	Difference
1	SUBTYPE subtype_name IS base_type	Supported.	-
2	SUBTYPE subtype_name IS base_type { precision [, scale] | RANGE low_value .. high_value } [NOT NULL]	Supported, with differences.	• In GaussDB, only the INT type supports the RANGE constraint. In Oracle Database, only PLS_INTEGER, BINARY_INTEGER, and their predefined subtypes support the RANGE constraint. • In GaussDB, the RANGE constraint cannot be specified when variables are defined. • In GaussDB, the character set information cannot be specified when the base type of subtype is the character data type.
3	SUBTYPE subtype_name IS base_type [NOT NULL]	Supported.	-

10.3 Control Statements

Table 10-8 Conditional statements

No.	Oracle Database	GaussDB
1	IF THEN	Supported.
2	IF THEN ELSE	Supported.
3	IF THEN ELSIF	Supported.

No.	Oracle Database	GaussDB
4	**simple CASE:** CASE selector WHEN selector_value_1 THEN statements_1 WHEN selector_value_2 THEN statements_2 ... WHEN selector_value_n THEN statements_n [ELSE else_statements END CASE;]	Supported.
5	**searched CASE:** CASE WHEN condition_1 THEN statements_1 WHEN condition_2 THEN statements_2 ... WHEN condition_n THEN statements_n [ELSE else_statements END CASE;]	Supported.

Table 10-9 LOOP statements

No.	Oracle Database	GaussDB
1	[label] LOOP statements END LOOP [label];	Supported.
2	EXIT;	Supported.
3	EXIT WHEN;	Supported.
4	CONTINUE;	Supported.
5	CONTINUE WHEN;	Supported.

Table 10-10 FOR LOOP statements

No.	Oracle Database	GaussDB	Difference
1	[label] FOR index IN [REVERSE] lower_bound..upper_bound LOOP statements END LOOP [label];	Supported, with differences.	When the keyword REVERSE is used in GaussDB, the lower bound must be greater than or equal to the upper bound; otherwise, the loop body is not executed.
2	EXIT WHEN;	Supported.	-
3	CONTINUE WHEN;	Supported.	-

Table 10-11 WHILE LOOP statement

No.	Oracle Database	GaussDB
1	[label] WHILE condition LOOP statements END LOOP [label];	Supported.

Table 10-12 GOTO statement

No.	Oracle Database	GaussDB
1	GOTO	Supported.

Table 10-13 NULL statement

No.	Oracle Database	GaussDB
1	NULL	Supported.

10.4 Collections and Records

Table 10-14 Types

No.	Oracle Database	GaussDB
1	Associative array (or index-by table)	Supported.
2	VARRAY (variable-size array)	Supported.

No.	Oracle Database	GaussDB
3	Nested table	Supported.
4	record	Supported.

Table 10-15 Syntax

No.	Oracle Database	GaussDB	Difference
1	Associative array (or index-by table) syntax: TABLE OF datatype [NOT NULL] INDEX BY { PLS_INTEGER | BINARY_INTEGER | VARCHAR2 (v_size) | data_type }	Supported, with differences.	• GaussDB does not support the PLS_INTEGER type. In GaussDB, the value of **data_type** can be a base data type or a record type, collection type, or array type defined in a stored procedure. The ref cursor type is not supported. • In GaussDB, NOT NULL does not take effect in the syntax. That is, the system does not check whether an element is NULL. • For details, see "Stored Procedure > Arrays, Collections, and Records > Collections" in *Developer Guide*.
2	VARRAY (variable-size array) syntax: { VARRAY | [VARYING] ARRAY } (size_limit) OF datatype [NOT NULL]	Supported, with differences.	• GaussDB does not support the NOT NULL syntax. • In GaussDB, **datatype** cannot be set to **varray** (varray cannot be nested). • To make the size_limit function take effect, enable the **varray_compat** parameter in the GUC parameter **behavior_compat_options**. • For details, see "Stored Procedure > Arrays, Collections, and Records > Arrays" in *Developer Guide*.
3	Nested table syntax: TABLE OF datatype [NOT NULL]	Supported, with differences.	• In GaussDB, NOT NULL does not take effect in the syntax. That is, the system does not check whether an element is NULL. • For details, see "Stored Procedure > Arrays, Collections, and Records > Collections" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
4	record syntax: TYPE record_type IS RECORD (field_definition [, field_definition]...) ;	Supported.	• Record columns can be defined as NOT NULL or a default value can be specified. If the record type is nested in other types, the default value and NOT NULL of the record type do not take effect. If a record variable is created using the package.record_type access type, the default value and NOT NULL of the record variable do not take effect. • For details, see "Stored Procedure > Arrays, Collections, and Records > Records" in *Developer Guide*.

Table 10-16 Constructor

No.	Oracle Database	GaussDB
1	collection_type ([value [, value]...])	Supported.

Table 10-17 Variable assignment

No.	Oracle Database	GaussDB	Difference
1	Associative array (or index-by table)	Supported.	-
2	VARRAY (variable-size array)	Supported, with differences.	• Values of different VARRAY data in GaussDB can be assigned to each other, depending on whether elements in the data can be implicitly converted to each other. • For details, see "Stored Procedure > Arrays, Collections, and Records > Arrays" in *Developer Guide*.
3	Nested table	Supported.	-

No.	Oracle Database	GaussDB	Difference
4	record	Supported, with differences.	• Values of different record data in GaussDB can be assigned to each other, depending on whether columns can be implicitly converted. • For details, see "Stored Procedure > Arrays, Collections, and Records > Records" in *Developer Guide*.

Table 10-18 Collection operators

No.	Oracle Database	GaussDB	Difference
1	=	Supported, with differences.	• Oracle Database: The sequence of collection members is ignored during comparison. • GaussDB: The comparison is performed strictly based on the sequence of collection members.
2	<>	Supported, with differences.	• Oracle Database: The sequence of collection members is ignored during comparison. • GaussDB: The comparison is performed strictly based on the sequence of collection members.
3	IS[NOT] NULL	Supported.	-
4	^=	Supported, with differences.	• Oracle Database: The sequence of collection members is ignored during comparison. • GaussDB: The comparison is performed strictly based on the sequence of collection members.
5	~=	Not supported.	-
6	IS[NOT] A SET	Not supported.	-
7	IS [NOT] EMPTY	Not supported.	-

No.	Oracle Database	GaussDB	Difference
8	expr [NOT] MEMBER [OF] nested_table	Not supported.	-
9	nested_table1 [NOT] SUBMULTISET [OF] nested_table2	Not supported.	-
10	[NOT] IN	Supported, with differences.	• Oracle Database: The sequence of collection members is ignored during comparison. • GaussDB: The comparison is performed strictly based on the sequence of collection members.

Table 10-19 MULTISET functions

No.	Oracle Database	GaussDB
1	MULTISET UNION [ALL | DISTINCT]	Supported.
2	MULTISET EXCEPT [ALL | DISTINCT]	Supported.
3	MULTISET INTERSECT [ALL | DISTINCT]	Supported.

Table 10-20 Collection type functions

No.	Oracle Database	GaussDB	Difference
1	exists(idx)	Supported.	-
2	extend[(count[, idx])]	Supported, with differences.	GaussDB supports only nested tables.
3	delete[(idx1[, idx2])]	Supported.	-
4	trim[(n)]	Supported, with differences.	GaussDB supports only nested tables.
5	count	Supported.	-

No.	Oracle Database	GaussDB	Difference
6	first	Supported.	-
7	last	Supported.	-
8	prior(idx)	Supported.	-
9	next(idx)	Supported.	-
10	limit	Supported, with differences.	GaussDB supports only nested tables.

Table 10-21 Record variable operations

No.	Oracle Database	GaussDB
1	Constructors	Supported.
2	%ROWTYPE to declare a variable	Supported.
3	Defining constants	Not supported.

Table 10-22 Collection-related functions

No.	Oracle Database	GaussDB	Difference
1	unnest_table(anynesttable)	Supported.	-
2	unnest_table(anyindexbytable)	Supported.	-
3	table(anyarray)	Not supported.	GaussDB uses the unnest (anyarray) function for equivalent rewriting.

10.5 Static SQL Statements

Table 10-23 Static query SQL statements

No.	Oracle Database	GaussDB	Difference
1	SELECT	Supported, with differences.	GaussDB and Oracle Database are different in some scenarios. GaussDB: Shared locks in different transactions do not block each other in the following scenarios: SELECT FOR SHARE - SELECT FOR SHARE; SELECT FOR SHARE - SELECT FOR KEY SHARE; SELECT FOR KEY SHARE - SELECT FOR KEY SHARE; SELECT FOR KEY SHARE - SELECT FOR NO KEY UPDATE; In the preceding scenarios, because there is no blocking between locks, the lock will not be skipped when SKIP LOCKED is specified for data that has non-blocking locks in other transactions.

Table 10-24 Static DML SQL statements

No.	Oracle Database	GaussDB	Difference
1	INSERT	Supported.	-
2	UPDATE	Supported.	-
3	DELETE	Supported.	-
4	MERGE	Supported.	-
5	LOCK TABLE	Supported.	-

No.	Oracle Database	GaussDB	Difference
6	INSERT ALL	Supported, with differences.	• Oracle Database does not support alias setting for the tables of into_clause, but GaussDB supports. • When into_clause specifies the sequence: – Oracle Database: If nextval is referenced for the first time, the next number of the current value is generated. Otherwise, the same number will always be returned. – GaussDB: The number generated by referencing nextval can increment automatically. • The plan_hint statement can take effect in Oracle Database but does not take effect in GaussDB.

Table 10-25 Static TCL SQL statements

No.	Oracle Database	GaussDB	Difference
1	COMMIT	Supported.	-
2	ROLLBACK	Supported.	-
3	SAVEPOINT	Supported.	-
4	SET TRANSACTION	Supported, with differences.	GaussDB does not support the NAME string and USE ROLLBACK SEGMENT rollback_segment syntax.

Table 10-26 Pseudocolumns

No.	Oracle Database	GaussDB	Difference
1	CURRVAL and NEXTVAL	Supported.	-
2	LEVEL	Supported.	-
3	OBJECT_VALUE	Not supported.	-

No.	Oracle Database	GaussDB	Difference
4	ROWID	Not supported.	-
5	ROWNUM	Supported, with differences.	It is not recommended that the ROWNUM condition be used in the JOIN ON clause. In GaussDB, when the ROWNUM condition is used in the JOIN ON clause, the behavior in the LEFT JOIN, RIGHT JOIN, FULL JOIN, and MERGE INTO scenarios is different from that in other databases, causing risks in service migration.

Table 10-27 Implicit cursor attributes

No.	Oracle Database	GaussDB	Difference
1	SQL%FOUND	Supported, with differences.	GaussDB does not update the implicit cursor result after COMMIT or ROLLBACK. Oracle Database updates the implicit cursor result after COMMIT or ROLLBACK.
2	SQL %NOTFOUND	Supported, with differences.	
3	SQL %ROWCOUNT	Supported, with differences.	
4	SQL%ISOPEN	Supported, with differences.	
5	SQL %BULK_ROWCOUNT	Not supported.	
6	SQL %BULK_EXCEPTIONS	Supported, with differences.	

Table 10-28 Explicit cursor syntax and keywords

No.	Oracle Database	GaussDB	Difference
1	CURSOR cursor_name [parameter_list] RETURN return_type;	Supported.	-
2	CURSOR cursor_name [parameter_list] [RETURN return_type] IS select_statement;	Supported.	-
3	OPEN	Supported.	-
4	CLOSE	Supported, with differences.	GaussDB is automatically closed in the exception, but Oracle Database is not automatically closed in the exception.
5	FETCH	Supported.	-
6	CURRENT OF CURSOR	Supported.	-

Table 10-29 Explicit cursor attributes

No.	Oracle Database	GaussDB
1	SQL%FOUND	Supported.
2	SQL%NOTFOUND	Supported.
3	SQL%ROWCOUNT	Supported.
4	SQL%ISOPEN	Supported.

Table 10-30 Cursor loop

No.	Oracle Database	GaussDB
1	FOR LOOP	Supported.

Table 10-31 Scenarios supported by autonomous transactions

No.	Oracle Database	GaussDB
1	Stored procedures	Supported.
2	Anonymous blocks	Supported.
3	Functions	Supported.
4	Packages	Supported.

10.6 Dynamic SQL Statements

Table 10-32 Dynamic SQL statement execution modes

No.	Oracle Database	GaussDB	Difference
1	EXECUTE IMMEDIATE	Supported, with differences.	• GaussDB uses the **dynamic_sql_compat** parameter to determines whether variables with the same name read the same parameter and check whether the input and output parameter types of the bound parameters are the same as those of the statement parameters when the stored procedure is called. • GaussDB does not support scenarios where some bound parameters in anonymous blocks are called. For example, when dynamic statements are nested in anonymous blocks, expressions are used to bind parameters. For details, see "Stored Procedure > Dynamic Statements > Dynamically Calling Anonymous Blocks" in *Developer Guide*. • GaussDB does not support RETURNING/RETURN INTO.
2	OPEN FOR, FETCH, CLOSE	Supported.	GaussDB uses the **dynamic_sql_compat** parameter to determines whether variables with the same name read the same parameter and check whether the input and output parameter types of the bound parameters are the same as those of the statement parameters when the stored procedure is called.

10.7 Triggers

Table 10-33 Trigger types

No.	Oracle Database	GaussDB	Difference
1	DML TRIGGER	Supported, with differences.	GaussDB: Compound DML triggers are not supported.
2	SYSTEM TRIGGER	Not supported.	-

Table 10-34 CREATE triggers

No.	Oracle Database	GaussDB	Difference
1	CREATE syntaxes: CREATE [OR REPLACE] [EDITIONABLE | NONEDITIONABLE] TRIGGER plsql_trigger_source	Supported, with differences.	GaussDB does not support EDITIONABLE | NONEDITIONABLE, but supports some behaviors of plsql_trigger_source.
2	plsql_trigger_source ::= syntax: [schema.] trigger_name [sharing_clause] [default_collation_clause] { simple_dml_trigger | instead_of_dml_trigger | compound_dml_trigger | system_trigger }	Supported, with differences.	GaussDB: The schema, sharing_clause, and default_collation_clause are not supported.

No.	Oracle Database	GaussDB	Difference
3	simple_dml_trigger ::= syntax: { BEFORE | AFTER } dml_event_clause [referencing_clause] [FOR EACH ROW] [trigger_edition_clause] [trigger_ordering_clause] [ENABLE | DISABLE] [WHEN (condition)] trigger_body	Supported, with differences.	GaussDB does not support referencing_clause, referencing_clause (instead, from referencing_table is used), trigger_edition_clause, trigger_ordering_clause, and ENABLE | DISABLE. It supports some behaviors of trigger_body. No error is reported in GaussDB when a statement-level BEFORE/AFTER TRIGGER is created in a view without INSTEAD OF TRIGGER. An error is reported when DML is executed.
4	dml_event_clause ::= syntax: { DELETE | INSERT | UPDATE [OF column [, column]...] } [OR { DELETE | INSERT | UPDATE [OF column [, column]...] }... ON [schema.] { table | view }	Not supported.	-
5	trigger_body ::= syntax: { plsql_block | CALL routine_clause }	Supported, with differences.	In GaussDB, plsql_block cannot be declared as PRAGMA AUTONOMOUS_TRANSACTION. For the second branch, similar syntax is supported. That is, a function is executed in EXECUTE PROCEDURE function_name (arguments); mode. The function needs to be defined by users. The function must be declared without parameters and the return type must be trigger. The function is executed when the trigger is enabled.

No.	Oracle Database	GaussDB	Difference
6	instead_of_dml_trigger ::= syntax: INSTEAD OF { DELETE | INSERT | UPDATE } [OR { DELETE | INSERT | UPDATE }]... ON [NESTED TABLE nested_table_column OF] [schema.] noneditioning_view [referencing_clause] [FOR EACH ROW] [trigger_edition_clause] [trigger_ordering_clause] [ENABLE | DISABLE] trigger_body	Supported, with differences.	GaussDB: The NESTED TABLE nested_table_column OF, referencing_clause, trigger_edition_clause, trigger_ordering_clause, and ENABLE | DISABLE are not supported.
7	compound_dml_trigger ::= syntax: CREATE trigger FOR dml_event_clause ON view COMPOUND TRIGGER INSTEAD OF EACH ROW IS BEGIN statement; END INSTEAD OF EACH ROW;	Not supported.	-
8	system_trigger ::= syntax: { BEFORE | AFTER | INSTEAD OF } { ddl_event [OR ddl_event]... | database_event [OR database_event]... } ON { [schema.] SCHEMA | [PLUGGABLE] DATABASE } [trigger_ordering_clause] [ENABLE | DISABLE] trigger_body	Not supported.	-

Table 10-35 ALTER trigger

No.	Oracle Database	GaussDB	Difference
1	ALTER TRIGGER [schema.] trigger_name { trigger_compile_clause | { ENABLE | DISABLE } | RENAME TO new_name | { EDITIONABLE | NONEDITIONABLE } } ;	Supported, with differences.	GaussDB: The schema, trigger_compile_clause, { ENABLE | DISABLE }, and { EDITIONABLE | NONEDITIONABLE } are not supported.

Table 10-36 drop trigger

No.	Oracle Database	GaussDB	Difference
1	DROP TRIGGER [schema.] trigger ;	Supported, with differences.	GaussDB does not support schemas. You need to add **ON table_name** to the end of **trigger_name**.

The *_TRIGGERS views in Oracle Database collect information about triggers. The views in GaussDB are different from those in Oracle Database. For details, see "DB_TRIGGERS", "ADM_TRIGGERS", "MY_TRIGGERSDB_TRIGGERS", "ADM_TRIGGERS", and "MY_TRIGGERS" in "System Catalogs and System Views > System Views > Other System Views" in *Developer Guide*.

Table 10-37 Compatibilities of nested, package, and standalone subprograms

No.	Oracle Database	GaussDB	Difference
1	Nested subprograms (subblocks)	Supported, with differences.	Overloading, definition of autonomous transactions, and SETOF are not supported. Only one qualifier can reference nested subprograms or variables of nested subprograms.
2	Package subprograms	Supported.	-
3	Standalone subprograms (including functions and procedures)	Supported.	-
4	Anonymous blocks	Supported.	-

Table 10-38 RETURN statements

No.	Oracle Database	GaussDB
1	Functions	Supported.
2	Procedures	Supported.
3	Anonymous blocks	Supported.

Table 10-39 Function-related parameters

No.	Oracle Database	GaussDB	Difference
1	DETERMINISTIC	Supported, with differences.	In GaussDB, it is IMMUTABLE.
2	PARALLEL_ENABLE	Not supported.	-
3	PIPELINED	Not supported.	-
4	RESULT_CACHE	Not supported.	-

Table 10-40 Parameter formats

No.	Oracle Database	GaussDB
1	IN	Supported.
2	OUT	Supported.
3	IN OUT	Supported.

Table 10-41 CREATE statements

No.	Oracle Database	GaussDB	Difference
1	CREATE FUNCTION	Supported, with differences.	GaussDB does not support the IF NOT EXISTS syntax. The sharing_clause is not supported. Only the invoker_rights_clause with the **function** attribute is supported. The keyword [EDITIONABLE | NONEDITIONABLE] is not supported. For details about the syntax, see "SQL Reference > SQL Syntax > C > CREATE FUNCTION" in *Developer Guide*.
2	CREATE LIBRARY	Not supported.	-
3	CREATE PACKAGE	Supported, with differences.	GaussDB does not support the IF NOT EXISTS syntax. The sharing_clause is not supported. Only the invoker_rights_clause with the **package** attribute is supported. The keyword [EDITIONABLE | NONEDITIONABLE] is not supported. For details about the syntax, see "SQL Reference > SQL Syntax > C > CREATE PACKAGE" in *Developer Guide*.
4	CREATE PACKAGE BODY	Supported, with differences.	GaussDB does not support the IF NOT EXISTS syntax. The sharing_clause and keyword [EDITIONABLE | NONEDITIONABLE] are not supported. For details about the syntax, see "SQL Reference > SQL Syntax > C > CREATE PACKAGE" in *Developer Guide*.
5	CREATE PROCEDURE	Supported, with differences.	GaussDB does not support the IF NOT EXISTS syntax. The sharing_clause and its following clauses are not supported. The keyword [EDITIONABLE | NONEDITIONABLE] is not supported. For details about the syntax, see "SQL Reference > SQL Syntax > C > CREATE PROCEDURE" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
6	CREATE TRIGGER	Supported, with differences.	For details about GaussDB syntax, see "SQL Reference > SQL Syntax > C > CREATE TRIGGER" in *Developer Guide*.
7	CREATE TYPE	Supported, with differences.	GaussDB does not support the varray, object type, and UNDER syntax. For details about the syntax, see "SQL Reference > SQL Syntax > C > CREATE TYPE" in *Developer Guide*.
8	CREATE TYPE BODY	Not supported.	-

Table 10-42 ALTER statements

No.	Oracle Database	GaussDB	Difference
1	ALTER FUNCTION	Supported, with differences.	GaussDB does not support keywords [EDITIONABLE | NONEDITIONABLE], REUSE, SETTINGS, and DEBUG. For details about the syntax, see "SQL Reference > SQL Syntax > A > ALTER FUNCTION" in *Developer Guide*.
2	ALTER LIBRARY	Not supported.	-
3	ALTER PACKAGE	Supported, with differences.	GaussDB does not support keywords [EDITIONABLE | NONEDITIONABLE], REUSE, SETTINGS, and DEBUG. For details about the syntax, see "SQL Reference > SQL Syntax > A > ALTER PACKAGE" in *Developer Guide*.
4	ALTER PROCEDURE	Supported, with differences.	GaussDB does not support keywords [EDITIONABLE | NONEDITIONABLE], REUSE, SETTINGS, and DEBUG. For details about the syntax, see "SQL Reference > SQL Syntax > A > ALTER PROCEDURE" in *Developer Guide*.
5	ALTER TRIGGER	Supported, with differences.	In GaussDB, only the trigger name can be modified. For details about the syntax, see "SQL Reference > SQL Syntax > A > ALTER TRIGGER" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
6	ALTER TYPE	Supported, with differences.	GaussDB supports only some statements. For details about the syntax, see "SQL Reference > SQL Syntax > A > ALTER TYPE" in *Developer Guide*.

Table 10-43 DROP statements

No.	Oracle Database	GaussDB	Difference
1	DROP FUNCTION	Supported.	-
2	DROP LIBRARY	Not supported.	-
3	DROP PACKAGE	Supported.	-
4	DROP PROCEDURE	Supported.	-
5	DROP TRIGGER	Supported, with differences.	The syntax in GaussDB is different. For details about the syntax, see "SQL Reference > SQL Syntax > D > DROP TRIGGER" in *Developer Guide*.
6	DROP TYPE	Supported, with differences.	GaussDB does not support keywords FORCE and VALIDATE. For details about the syntax, see "SQL Reference > SQL Syntax > D > DROP TYPE" in *Developer Guide*.
7	DROP TYPE BODY	Not supported.	-

Table 10-44 Keywords related to functions, procedures, and anonymous blocks

No.	Oracle Database	GaussDB	Difference
1	ACCESSIBLE BY	Not supported.	-

No.	Oracle Database	GaussDB	Difference
2	AGGREGATE	Supported, with differences.	• GaussDB does not support Oracle Database's aggregate using [schema.] implementation_type. • For details about GaussDB syntax, see "SQL Reference > SQL Syntax > C > CREATE AGGREGATE" in *Developer Guide*. The syntax is different, but the implementation functions are the same.
3	DETERMINISTIC	Supported, with differences.	GaussDB supports the keyword DETERMINISTIC only in syntax, but does not support this function.
4	PIPE ROW	Not supported.	-
5	PIPELINED	Not supported.	-
6	SQL_MACRO	Not supported.	-
7	RESTRICT_REFERENCES	Not supported.	-
8	INLINE	Not supported.	-

Table 10-45 Keywords related to exception handling

No.	Oracle Database	GaussDB	Difference
1	EXCEPTION_INIT	Supported, with differences.	Binding with system error codes is not supported in GaussDB.
2	Exception	Supported.	-
3	Exception Handler	Supported.	-
4	SQLCODE	Supported.	-
5	SQLERRM	Supported.	-

11 System Functions

Compatible functions are classified into single-row functions, user-defined functions, aggregate functions, analytic functions, object reference functions, model functions, and OLAP functions.

11.1 Single-Row Functions

No.	Oracle Database	GaussDB
1	Numeric functions	Supported, with differences.
2	Character functions returning character values	Supported, with differences.
3	Character functions returning number values	Supported, with differences.
4	Character set functions	Not supported.
5	Collation functions	Not supported.
6	Datetime functions	Supported, with differences.
7	General comparison functions	Supported, with differences.
8	Conversion functions	Supported, with differences.
9	Large object functions	Supported, with differences.
10	Collection functions	Not supported.
11	Hierarchical functions	Supported.
12	Data mining functions	Not supported.
13	XML functions	Supported, with differences.
14	JSON functions	Not supported.

No.	Oracle Database	GaussDB
15	**Encoding and decoding functions**	Supported, with differences.
16	**Null-related functions**	Supported.
17	**Environment and identifier functions**	Supported, with differences.

Table 11-1 Numeric functions

No.	Oracle Database	GaussDB	Difference
1	ABS	Supported.	-
2	ACOS	Supported.	-
3	ASIN	Supported.	-
4	ATAN	Supported.	-
5	ATAN2	Supported.	-
6	BITAND	Supported.	-
7	CEIL	Supported.	-
8	COS	Supported.	-
9	COSH	Supported.	-
10	EXP	Supported.	-
11	FLOOR	Supported.	-
12	LN	Supported.	-
13	LOG	Supported.	-

No.	Oracle Database	GaussDB	Difference
14	MOD	Supported, with differences.	• The return types are different. In Oracle Database, the return types include BINARY_DOUBLE, BINARY_FLOAT, and NUMBER. In GaussDB, the return types include int2, int4, int8, and numeric. • If the first input parameter is of the numeric type, the second parameter must be of the int, numeric, or a type that can be converted to numeric. If **a_format_version** is set to **10c**, **a_format_dev_version** is set to **s6**, and the first parameter is of the text type that can be converted to numeric, the second parameter must be of the int4 type or a type with the value range smaller than int4.
15	NANVL	Supported, with differences.	GaussDB: NaN cannot be obtained by directly declaring or dividing a floating-point number by 0.
16	POWER	Supported.	-
17	REMAINDER	Supported, with differences.	The data types of the returned values are different. GaussDB: • If one input value is of the float4 type and the other is of the numeric type, values of the float4 type are returned. • If both input values are of the float4 type, values of the float4 type are returned. • If both input values are of the float8 type, values of the float8 type are returned. • For other data types, values of the numeric type are returned. Oracle Database: The type of returned values is number.

No.	Oracle Database	GaussDB	Difference
18	ROUND	Supported, with differences.	• For the first parameter **n** of the float type, GaussDB has precision loss, and the precision is lower than that of Oracle Database. • The returned types are different. For round(n, integer), Oracle Database returns the NUMBER type, and GaussDB returns the numeric type. For round(n), Oracle Database returns the data type of **n**, and GaussDB returns only the float8 and numeric types. The float4 return type is missing. • The logic for the GaussDB to determine that the input parameter is null and the execution framework to return null is different from that in Oracle Database. `SELECT round(NULL,'q');` Oracle Database reports null, and GaussDB reports the error: invalid input syntax for integer: "q".
19	SIGN	Supported.	-
20	SIN	Supported.	-
21	SINH	Supported.	-
22	SQRT	Supported.	-
23	TAN	Supported.	-
24	TANH	Supported, with differences.	The data types of the returned values are different. • GaussDB: – If an input value is of the float8 type, a value of the float8 type is returned. – If an input value is of the numeric type, a value of the numeric type is returned. • Oracle: The type of returned values is number.
25	TRUNC	Supported.	-
26	WIDTH_BUCKET	Supported.	-

Table 11-2 Character functions returning character values

No.	Oracle Database	GaussDB	Difference
1	CHR	Supported, with differences.	• If the entered number does not comply with the existing character set, GaussDB reports an error in JDBC and Oracle Database returns garbled characters. • If you enter **0** or **256**, Oracle Database returns characters whose ASCII code is 0, and GaussDB truncates the characters at **'\0;**.
2	CONCAT	Supported.	-
3	INITCAP	Supported, with differences.	The returned value is restricted by the database character set. As a result, the returned result is different from that in Oracle Database.
4	LOWER	Supported, with differences.	• The types of returned values are different. The data types of Oracle Database are the same as the input types. • The time format is implicitly converted. When the time type is entered, the time type is implicitly converted to a character string and then the lower operation is performed. SELECT LOWER(TO_DATE('2012-12-10','YYYY-MM-DD')); Oracle Database returns 10-DEC-12, and GaussDB returns 2012-12-10 00:00:00. • The returned value is restricted by the database character set. As a result, the returned result is different from that in Oracle Database.
5	LPAD	Supported.	-
6	LTRIM	Supported, with differences.	The return value types are different. If the input is of the character data type, Oracle Database returns the VARCHAR2 type. If the input is of the national character set specified during database creation, Oracle Database returns the NVARCHAR2 type. If the input is of the LOB type, Oracle Database returns the LOB type. GaussDB returns the TEXT type.

No.	Oracle Database	GaussDB	Difference
7	NCHR	Supported, with differences.	• The byte length of the returned value is different from that of Oracle Database. • The returned value is restricted by the database character set. As a result, the returned result is different from that in Oracle Database. • When the byte array corresponding to the input parameter is returned, a question mark (?) is returned if a single byte ranges from 0x80 to 0xFF. In Oracle Database, a question mark (?) is returned, no output is returned, or an error is reported.
8	NLS_LOWER	Supported, with differences.	• The return value types are different. If the input is of the character data type, Oracle Database returns the VARCHAR2 type. If the input is of the LOB type, Oracle Database returns the LOB type. GaussDB returns the TEXT type. • In Oracle Database, the **nlsparam** parameter can be of a type except nls_sort, and no error is reported. GaussDB supports only nls_sort. • The returned value is restricted by the database character set. As a result, the returned result is different from that in Oracle Database.
9	NLS_UPPER	Supported, with differences.	• The return value types are different. If the input is of the character data type, Oracle Database returns the VARCHAR2 type. If the input is of the LOB type, Oracle Database returns the LOB type. GaussDB returns the TEXT type. • In Oracle Database, the **nlsparam** parameter can be of a type except nls_sort, and no error is reported. GaussDB supports only nls_sort. • The returned value is restricted by the database character set. As a result, the returned result is different from that in Oracle Database.
10	NLSSORT	Supported.	-

No.	Oracle Database	GaussDB	Difference

11	REGEXP_REPLACE	Supported, with differences.	• GaussDB input parameter **source_char** does not support the NCLOB type.
			• The meaning of the **'n'** option in the **match_param** input parameter is different. In GaussDB, the **'n'** option has the same meaning as the **'m'** option, indicating that the multi-row matching mode is used. In Oracle Database, it indicates that the dot (.) can match the '\n' character. If this option is not specified, the '\n' character cannot be matched by default. In GaussDB, the dot (.) matches '\n' by default. You do not need to specify the option.
			• The matching results of some regular expressions may be different.
			`SELECT REGEXP_REPLACE('abc01234xyz', '(.*?)(\d+)(.*)', '#', 'g') FROM DUAL;`
			Oracle Database reports an error, and GaussDB returns **#####xyz**.
			• The matching results may be different when Chinese characters are entered in the UTF-8 character set. Oracle needs to implement regular expression matching for Chinese character strings in the GBK character set.
			• The matching results of regular expressions that contain some escape characters may be different.
			`SELECT REGEXP_REPLACE('abcabc', '\abc', '#', 'g') FROM DUAL;`
			Oracle Database reports an error, and GaussDB returns **abcabc**.
			• The matching rules are affected by the **aformat_regexp_match** parameter. For details about the affected specifications, see the REGEXP_REPLACE function in "SQL Reference > Functions and Operators > Character Processing Functions and Operators" in *Developer Guide*.

12	REGEXP_SUBSTR	Supported, with differences.	The matching rules are affected by the **aformat_regexp_match** parameter. For details about the affected specifications, see the REGEXP_SUBSTR function in "SQL Reference > Functions and Operators > Character Processing Functions and Operators" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
13	REPLACE	Supported.	-
14	RPAD	Supported.	-
15	RTRIM	Supported.	-
16	SUBSTR	Supported.	-
17	TRANSLATE	Supported.	-
18	TRIM	Supported.	-
19	UPPER	Supported, with differences.	• The types of returned values are different. The data types of Oracle Database are the same as the input types. GaussDB returns the TEXT type. • The time format is implicitly converted. When the time type is entered, the time type is implicitly converted to a character string and then the upper operation is performed. `SELECT UPPER(TO_DATE('2012-12-10','YYYY-MM-DD'));` Oracle Database returns 10-DEC-12, and GaussDB returns 2012-12-10 00:00:00. • The returned value is restricted by the database character set. As a result, the returned result is different from that in Oracle Database.
20	INSTRB	Supported.	-

Table 11-3 Character functions returning number values

No.	Oracle Database	GaussDB	Difference
1	ASCII	Supported, with differences.	The types of returned values are different. Oracle Database returns the uint4 type, while GaussDB returns the int4 type.
2	INSTR	Supported.	-
3	LENGTH	Supported.	-

No.	Oracle Database	GaussDB	Difference
4	REGEXP_COUNT	Supported, with differences.	• GaussDB input parameter **source_char** does not support the NCLOB type. • The meaning of the **'n'** option in the **match_param** input parameter is different. In GaussDB, the **'n'** option has the same meaning as the **'m'** option, indicating that the multi-row matching mode is used. In Oracle Database, it indicates that the dot (.) can match the '\n' character. If this option is not specified, the '\n' character cannot be matched by default. In GaussDB, the dot (.) matches '\n' by default. You do not need to specify the option. • The matching results of some regular expressions may be different. • The matching results may be different when Chinese characters are entered in the UTF-8 character set. Oracle needs to implement regular expression matching for Chinese character strings in the GBK character set. • The matching results of regular expressions that contain some escape characters may be different. • The matching rules are affected by the **aformat_regexp_match** parameter. For details about the affected specifications, see the REGEXP_COUNT function in "SQL Reference > Functions and Operators > Character Processing Functions and Operators" in *Developer Guide*.
5	REGEXP_INSTR	Supported, with differences.	The matching rules are affected by the **aformat_regexp_match** parameter. For details about the affected specifications, see the REGEXP_INSTR function in "SQL Reference > Functions and Operators > Character Processing Functions and Operators" in *Developer Guide*.
6	LENGTHC	Supported.	-

Table 11-4 Datetime functions

No.	Oracle Database	GaussDB	Difference
1	ADD_MONTHS	Supported, with differences.	• From A.D. to B.C., the difference between GaussDB and Oracle Database is one year. • The earliest year can be –4714 in GaussDB and –4713 in Oracle Database.
2	CURRENT_DATE	Supported, with differences.	GaussDB: The **nls_date_format** parameter cannot be used to set the time display format.
3	CURRENT_TIMESTAMP	Supported, with differences.	The value ranges from 0 to 9 in Oracle Database. The value ranges from 0 to 6 in GaussDB. The trailing zeros in microseconds are not displayed.
4	DBTIMEZONE	Supported, with differences.	GaussDB: The timestamp API with the built-in tz cannot be called.
5	EXTRACT	Supported.	-
6	LAST_DAY	Supported, with differences.	The types of returned values are different. The return value type in GaussDB is timestamp without time zone, and that in Oracle Database is date.
7	LOCALTIMESTAMP	Supported, with differences.	The value ranges from 0 to 9 in Oracle Database. The value ranges from 0 to 6 in GaussDB. The trailing zeros in microseconds are not displayed.
8	MONTHS_BETWEEN	Supported, with differences.	The input parameter types are different. GaussDB input parameters are of the timestamp without time zone type, and Oracle input parameters are of the date type.
9	NEW_TIME	Supported, with differences.	When the first input parameter of the new_time function is a literal, the literal format and the return value type of the function are different from those in Oracle Database.
10	NEXT_DAY	Supported.	-

No.	Oracle Database	GaussDB	Difference
11	NUMTODSINTERVAL	Supported, with differences.	GaussDB: The dsinterval type is not supported. Currently, interval is used to be compatible with the dsinterval type.
12	NUMTOYMINTERVAL	Supported, with differences.	GaussDB: The yminterval type is not supported. Currently, interval is used to be compatible with the yminterval type.
13	SESSIONTIMEZONE	Supported, with differences.	• Difference in the value assignment syntax: In GaussDB, the SET SESSION TIME ZONE 8 syntax is used. In Oracle Database, alter session set time_zone= '+08:00' is used. • Default value difference: The time zone name is used in GaussDB, for example, **PRC**. The offset is used in Oracle Database, for example, **+08:00**.
14	SYS_EXTRACT_UTC	Supported.	-
15	SYSDATE	Supported, with differences.	The types of returned values are different. The return value type in GaussDB is timestamp without time zone, and that in Oracle Database is date.
16	SYSTIMESTAMP	Supported, with differences.	GaussDB supports only six digits for millisecond calculation, and Oracle Database supports nine digits.
17	TO_CHAR	Supported, with differences.	The fmt '5' is not included in the Oracle Database documents and is not adapted.
18	TO_DSINTERVAL	Supported, with differences.	GaussDB: The dsinterval type is not supported. Currently, interval is used to be compatible with the dsinterval type.
19	TO_TIMESTAMP	Supported, with differences.	GaussDB supports only six digits for millisecond calculation, and Oracle Database supports nine digits.
20	TO_TIMESTAMP_TZ	Supported, with differences.	The timestamptz of GaussDB is equivalent to the timestampwithloacltimezone of Oracle Database. The type corresponding to timestamptz of Oracle Database is missing. The value of **nls_date_language** can only be **ENGLISH** or **AMERICAN**.
21	TO_YMINTERVAL	Supported, with differences.	GaussDB: The yminterval type is not supported. Currently, interval is used to be compatible with the yminterval type.

No.	Oracle Database	GaussDB	Difference
22	TRUNC	Supported, with differences.	The type returned by GaussDB is the same as the type of the first input parameter. Oracle Database always returns the date type. In addition, the supported formats are different in the two databases. For details about the supported formats, see "SQL Reference > Functions and Operators > Date and Time Processing Functions and Operators" in *Developer Guide*.
23	TZ_OFFSET	Supported, with differences.	When a time zone name is received as an input parameter, the types of the time zone name are less than those of Oracle Database.

Table 11-5 General comparison functions

No.	Oracle Database	GaussDB	Difference
1	GREATEST	Supported, with differences.	• GaussDB: The comparison mode specified by the **NLS_SORT** parameter is not supported. Only binary comparison is supported. • GaussDB: Expressions in multiple languages are not supported.
2	LEAST	Supported, with differences.	• GaussDB: The comparison mode specified by the **NLS_SORT** parameter is not supported. Only binary comparison is supported. • GaussDB: Expressions in multiple languages are not supported.

Table 11-6 Conversion functions

No.	Oracle Database	GaussDB	Difference
1	ASCIISTR	Supported.	-
2	CAST	Supported, with differences.	• GaussDB: The MULTISET clause is not supported. • GaussDB: The **nlsparam** parameter is not supported.

No.	Oracle Database	GaussDB	Difference
3	HEXTORAW	Supported.	-
4	RAWTOHEX	Supported.	-
5	TO_BINARY_DOUBLE	Supported, with differences.	GaussDB: The **nlsparam** parameter is not supported.
6	TO_BINARY_FLOAT	Supported, with differences.	GaussDB: The **nlsparam** parameter is not supported.
7	TO_BLOB	Supported, with differences.	• GaussDB: The long raw type is not supported. • GaussDB: The bfile and mime_type types are not supported.
8	TO_CLOB	Supported.	-
9	TO_DATE	Supported, with differences.	• Multi-language parameters are not supported. • The returned types are different. • The control parameter **NLS_DATE_FORMAT** is missing. • Some formats are not supported. • fmt = 'j'. The output before October 15, 1582 in Oracle Database is inconsistent with that in GaussDB. • If there is no separator, the value may be different from that in Oracle Database. Take to_date('220725','yymmdd') as an example. If yy/rr is parsed based on the fixed length 4, the year is parsed as 2207 and the month is parsed as 25. The month 25 is invalid, and an error will be reported.
10	TO_MULTI_BYTE	Supported.	-
11	TO_NCHAR	Supported, with differences.	GaussDB: The input parameter type is converted to text. Oracle Database: The input parameter type is converted to the national character set.

No.	Oracle Database	GaussDB	Difference
12	TO_NUMBER	Supported, with differences.	GaussDB does not support the **NLS_PARAM** parameter. The differences between the fmt options of GaussDB and Oracle Database are as follows: 1. $ GaussDB does not support this fmt. 2. Comma (,) GaussDB: Commas (,) can appear at any position of fmt. Oracle Database: ● In format, commas can only appear in the integer part and cannot appear at the beginning of a number. In the original data, commas can appear at the beginning of a number. ● The number and position of commas in the format can be different from those in the original data, but the position of the last comma must be the same. ● Consecutive commas in the original data and the format are equivalent to no comma. ● If the original data does not contain commas, the number of digits after the last comma in the format must be the same as that in the original data. 3. B GaussDB does not support this function. 4. C GaussDB does not support the **NLS** parameter. 5. G GaussDB does not support the **NLS** parameter. 6. L GaussDB does not support the **NLS** parameter. 7. U GaussDB does not support the **NLS** parameter. 8. D

No.	Oracle Database	GaussDB	Difference
			GaussDB does not support the **NLS** parameter. 9. PR GaussDB: It is equivalent to S. A negative number is returned. Oracle Database: Returns the negative value in the angle brackets (< >). Returns a positive value with leading and trailing spaces. Restriction: PR format elements can only appear at the last position of the digital format model. 10. RN | rn GaussDB does not support this function. TM| TM9 | TMe GaussDB does not support this function. 11. V GaussDB does not support this function. 12. FM When the fm exists in GaussDB, the comma in the format can be more than that in the original data. Otherwise, the comma must be the same. In Oracle Database, spaces before and after the return value are retained. 13. EEEE GaussDB does not support this function.
13	TO_SINGLE_BYTE	Supported.	-
14	TREAT	Supported, with differences.	GaussDB: The period (.) operator cannot be used to obtain values, and the values cannot be converted to the object type.
15	UNISTR	Supported, with differences.	GaussDB supports only UTF-8 encoding. Oracle Database supports UTF-8 and UTF-16 encodings.

Table 11-7 Large object functions

No.	Oracle Database	GaussDB	Difference
1	EMPTY_BLOB	Supported.	-
2	EMPTY_CLOB	Supported, with differences.	The CLOB type in GaussDB does not support the locator concept in Oracle Database.

Table 11-8 Hierarchical functions

No.	Oracle Database	GaussDB	Difference
1	SYS_CONNECT_BY_PATH	Supported, with differences.	In GaussDB, the column specified by the first input parameter can only be of the CHAR, VARCHAR, NVARCHAR2, TEXT, INT1, INT2, INT4, INT8, FLOAT4, FLOAT8, NUMERIC type. Currently, the **col** input of the function cannot be an expression. If the column content is the same as the separator, no error is reported.

Table 11-9 XML functions

No.	Oracle Database	GaussDB	Difference
1	EXISTSNODE	Supported, with differences.	If the input parameter has a namespace, aliases must be defined for both the XPath and namespace.
2	EXTRACTVALUE	Supported, with differences.	Currently, only XPath 1.0 is supported.
3	SYS_XMLAGG	Supported, with differences.	This is an alias of xmlagg and can be replaced with xmlagg.
4	XMLAGG	Supported.	-
5	XMLCOMMENT	Supported.	-
6	XMLCONCAT	Supported.	-

No.	Oracle Database	GaussDB	Difference
7	XMLELEMENT	Supported, with differences.	For xmlelement and xmlattributes, when the value of **name** is **NULL**, the database behavior is different from that in Oracle Database. When the **name** column of **xmlelement** is set to **NULL**, the name information is empty and the attribute information is not displayed. When the **name** column of **xmlattributes** is set to **NULL**, the attribute information is not displayed.
8	XMLEXISTS	Supported, with differences.	GaussDB input parameter is of the XML type.
9	XMLFOREST	Supported, with differences.	GaussDB return value is of the XML type. GaussDB does not support the EVALNAME syntax.
10	XMLPARSE	Supported, with differences.	GaussDB return value is of the XML type. GaussDB does not support the WELLFORMED syntax.
11	XMLROOT	Supported, with differences.	GaussDB return value is of the XML type.
12	JSON_OBJECT	Supported.	-
13	XMLTABLE	Supported, with differences.	GaussDB: The XPath 1.0 expression is used to select data from the XML file. The default namespace cannot be declared, multiple groups of inputs and aliases cannot be obtained, the passing_clause clause of the input data cannot be omitted, and the RETURNING SEQUENCE BY REF and (SEQUENCE) BY REF clauses are not supported.
14	GETSTRINGVAL	Supported.	-
15	GETCLOBVAL	Supported.	-
16	XMLSEQUENCE	Supported.	-

Table 11-10 Encoding and decoding functions

No.	Oracle Database	GaussDB	Difference
1	DECODE	Supported.	-
2	DUMP	Supported, with differences.	The returned results of the numeric and time types in GaussDB are inconsistent with those in Oracle Database due to different storage formats. For example: ● In GaussDB, SELECT dump(123); returns **Typ=23 Len=4: 123,0,0,0**. ● In Oracle Database, SELECT dump(123) FROM dual; returns **Typ=2 Len=3: 194,2,24**.
3	ORA_HASH	Supported, with differences.	GaussDB has the following behaviors: ● The input parameter of the time type is converted into the character string type and then hashed. ● The **maxbucket** parameter is not supported.
4	VSIZE	Supported, with differences.	The returned results of the numeric and time types in GaussDB are inconsistent with those in Oracle Database due to different storage formats. For example: ● In GaussDB, SELECT vsize(999); returns **4**. ● In Oracle Database, SELECT vsize(999) FROM dual; returns **3**.

Table 11-11 Null-related functions

No.	Oracle Database	GaussDB
1	COALESCE	Supported.
2	LNNVL	Supported.
3	NULLIF	Supported.
4	NVL	Supported.
5	NVL2	Supported.

Table 11-12 Environment and identifier functions

No.	Oracle Database	GaussDB	Difference
1	SYS_CONTEXT	Supported, with differences.	GaussDB returns NULL for unsupported parameters. The following parameters are not supported: 'action' 'is_application_root' 'is_application_pdb' 'audited_cursorid' 'authenticated_identity' 'authentication_data' 'authentication_method' 'cdb_domain' 'cdb_name' 'client_identifier' 'con_id' 'con_name' 'current_sql_length' 'db_domain' 'db_supplemental_log_level' 'dblink_info' 'drain_status' 'entryid' 'enterprise_identity' 'fg_job_id' 'global_uid' 'identification_type' 'instance' 'is_dg_rolling_upgrade' 'ldap_server_type' 'module' 'network_protocol' 'nls_calendar' 'nls_sort' 'nls_territory' 'oracle_home' 'os_user' 'platform_slash'

No.	Oracle Database	GaussDB	Difference
			'policy_invoker' 'proxy_enterprise_identity' 'proxy_user' 'proxy_userid' 'scheduler_job' 'session_edition_id' 'session_edition_name' 'sessionid' 'statementid' 'terminal' 'unified_audit_sessionid' 'session_default_collation' 'client_info' 'bg_job_id' 'client_program_name' 'current_bind' 'global_context_memory' 'host' 'current_sqln'
2	SYS_GUID	Supported.	-
3	USER	Supported, with differences.	The return value types are inconsistent. The return value type of GaussDB is name, and that of Oracle Database is varchar2.

No.	Oracle Database	GaussDB	Difference
4	USERENV	Supported, with differences.	GaussDB returns NULL for unsupported parameters. The following parameters are not supported: • 'action' • 'is_application_root' • 'is_application_pdb' • 'audited_cursorid' • 'authenticated_identity' • 'authentication_data' • 'authentication_method' • 'cdb_domain' • 'cdb_name' • 'client_identifier' • 'con_id' • 'con_name' • 'current_sql_length' • 'db_domain' • 'db_supplemental_log_level' • 'dblink_info' • 'drain_status' • 'entryid' • 'enterprise_identity' • 'fg_job_id' • 'global_uid' • 'identification_type' • 'is_dg_rolling_upgrade' • 'ldap_server_type' • 'module' • 'network_protocol' • 'nls_calendar' • 'nls_sort' • 'nls_territory' • 'oracle_home' • 'os_user' • 'platform_slash' • 'policy_invoker' • 'proxy_enterprise_identity'

No.	Oracle Database	GaussDB	Difference
			• 'proxy_user' • 'proxy_userid' • 'scheduler_job' • 'session_edition_id' • 'session_edition_name' • 'sessionid' • 'statementid' • 'terminal' • 'unified_audit_sessionid' • 'session_default_collation' • 'client_info' • 'bg_job_id' • 'client_program_name' • 'current_bind' • 'global_context_memory' • 'host' • 'current_sqln'

11.2 Other Functions

No.	Oracle Database	GaussDB
1	Aggregate functions	Supported.
2	Analytic functions	Supported.
3	Object reference functions	Not supported.
4	Models functions	Not supported.
5	OLAP functions	Not supported.
6	Data cartridge functions	Not supported.
7	User-defined functions	Supported.

Table 11-13 Aggregate functions

No.	Oracle Database	GaussDB	Difference
1	AVG	Supported.	-

No.	Oracle Database	GaussDB	Difference
2	CORR	Supported.	-
3	COUNT	Supported.	-
4	COVAR_POP	Supported.	-
5	COVAR_SAMP	Supported.	-
6	CUME_DIST	Supported.	-
7	DENSE_RANK	Supported.	-
8	FIRST	Supported.	The KEEP syntax used in GaussDB is compatible with Oracle Database.
9	GROUPING	Supported.	-
10	LAST	Supported.	The KEEP syntax used in GaussDB is compatible with Oracle Database.
11	LISTAGG	Supported.	-
12	MAX	Supported.	-
13	MEDIAN	Supported.	-
14	MIN	Supported.	-
15	PERCENT_RANK	Supported.	-
16	PERCENTILE_CONT	Supported.	-
17	RANK	Supported.	-
18	REGR_ (Linear Regression)	Supported.	-
19	STDDEV	Supported.	-
20	STDDEV_POP	Supported.	-
21	STDDEV_SAMP	Supported.	-
22	SUM	Supported.	-
23	VAR_POP	Supported.	-
24	VAR_SAMP	Supported.	-
25	VARIANCE	Supported.	-

No.	Oracle Database	GaussDB	Difference
26	WM_CONCAT	Supported.	-

Table 11-14 Analytic functions

No.	Oracle Database	GaussDB	Difference
1	FIRST_VALUE	Supported.	-
2	LAG	Supported.	-
3	LAST_VALUE	Supported.	-
4	LEAD	Supported.	-
5	NTH_VALUE	Supported, with differences.	• Oracle Database: The FROM FIRST|LAST syntax format is supported. • GaussDB: The FROM FIRST|LAST syntax format is not supported.
6	NTILE	Supported.	-
7	ROW_NUMBER	Supported.	-
8	RATIO_TO_REPORT	Supported.	-

12 System Views

GaussDB is compatible with some Oracle Database system views, as shown in the following table.

For details about columns in system views, see "System Views" in *Developer Guide*.

Table 12-1 Supported views

No.	Oracle Database	GaussDB
1	ALL_ALL_TABLES	DB_ALL_TABLES
2	ALL_COL_PRIVS	DB_COL_PRIVS
3	ALL_COLL_TYPES	DB_COLL_TYPES
4	ALL_ERRORS	DB_ERRORS
5	ALL_IND_COLUMNS	DB_IND_COLUMNS
6	ALL_COL_COMMENTS	DB_COL_COMMENTS
7	ALL_CONS_COLUMNS	DB_CONS_COLUMNS
8	ALL_CONSTRAINTS	DB_CONSTRAINTS
9	ALL_DEPENDENCIES	DB_DEPENDENCIES
10	ALL_DIRECTORIES	DB_DIRECTORIES
11	ALL_IND_EXPRESSIONS	DB_IND_EXPRESSIONS
12	ALL_IND_PARTITIONS	DB_IND_PARTITIONS
13	ALL_INDEXES	DB_INDEXES
14	ALL_IND_SUBPARTITIONS	DB_IND_SUBPARTITIONS
15	ALL_OBJECTS	DB_OBJECTS
16	ALL_PART_COL_STATISTICS	DB_PART_COL_STATISTICS

No.	Oracle Database	GaussDB
17	ALL_PART_KEY_COLUMNS	DB_PART_KEY_COLUMNS
18	ALL_PART_TABLES	DB_PART_TABLES
19	ALL_SCHEDULER_JOB_ARGS	DB_SCHEDULER_JOB_ARGS
20	ALL_SCHEDULER_PROGRAM_ARGS	DB_SCHEDULER_PROGRAM_ARGS
21	ALL_SEQUENCES	DB_SEQUENCES
22	ALL_SUBPART_COL_STATISTICS	DB_SUBPART_COL_STATISTICS
23	ALL_SUBPART_KEY_COLUMNS	DB_SUBPART_KEY_COLUMNS
24	ALL_SYNONYMS	DB_SYNONYMS
25	ALL_TAB_COL_STATISTICS	DB_TAB_COL_STATISTICS
26	ALL_TAB_COMMENTS	DB_TAB_COMMENTS
27	ALL_TAB_HISTOGRAMS	DB_TAB_HISTOGRAMS
28	ALL_TAB_STATS_HISTORY	DB_TAB_STATS_HISTORY
29	ALL_TYPES	DB_TYPES
30	ALL_PART_INDEXES	DB_PART_INDEXES
31	ALL_PROCEDURES	DB_PROCEDURES
32	ALL_SOURCE	DB_SOURCE
33	ALL_TAB_COLUMNS	DB_TAB_COLUMNS
34	ALL_TAB_PARTITIONS	DB_TAB_PARTITIONS
35	ALL_TAB_SUBPARTITIONS	DB_TAB_SUBPARTITIONS
36	ALL_TABLES	DB_TABLES
37	ALL_TRIGGERS	DB_TRIGGERS
38	ALL_USERS	DB_USERS
39	ALL_VIEWS	DB_VIEWS
40	DBA_AUDIT_OBJECT	ADM_AUDIT_OBJECT
41	DBA_AUDIT_SESSION	ADM_AUDIT_SESSION
42	DBA_AUDIT_STATEMENT	ADM_AUDIT_STATEMENT
43	DBA_AUDIT_TRAIL	ADM_AUDIT_TRAIL
44	DBA_COL_COMMENTS	ADM_COL_COMMENTS

No.	Oracle Database	GaussDB
45	DBA_COL_PRIVS	ADM_COL_PRIVS
46	DBA_COLL_TYPES	ADM_COLL_TYPES
47	DBA_ARGUMENTS	ADM_ARGUMENTS
48	DBA_CONSTRAINTS	ADM_CONSTRAINTS
49	DBA_DATA_FILES	ADM_DATA_FILES
50	DBA_CONS_COLUMNS	ADM_CONS_COLUMNS
51	DBA_DEPENDENCIES	ADM_DEPENDENCIES
52	DBA_DIRECTORIES	ADM_DIRECTORIES
53	DBA_PART_COL_STATISTICS	ADM_PART_COL_STATISTICS
54	DBA_PART_TABLES	ADM_PART_TABLES
55	DBA_RECYCLEBIN	ADM_RECYCLEBIN
56	DBA_ROLE_PRIVS	ADM_ROLE_PRIVS
57	DBA_ROLES	ADM_ROLES
58	DBA_SCHEDULER_JOB_ARGS	ADM_SCHEDULER_JOB_ARGS
59	DBA_SCHEDULER_PROGRAMS	ADM_SCHEDULER_PROGRAMS
60	DBA_SCHEDULER_PROGRAM_ARGS	ADM_SCHEDULER_PROGRAM_ARGS
61	DBA_HIST_SNAPSHOT	ADM_HIST_SNAPSHOT
62	DBA_HIST_SQL_PLAN	ADM_HIST_SQL_PLAN
63	DBA_HIST_SQLSTAT	ADM_HIST_SQLSTAT
64	DBA_HIST_SQLTEXT	ADM_HIST_SQLTEXT
65	DBA_ILMDATAMOVEMENT-POLICIES	GS_ADM_ILMDATAMOVEMENTPOLICIES
66	DBA_ILMEVALUATIONDE-TAILS	GS_ADM_ILMEVALUATIONDETAILS
67	DBA_ILMOBJECTS	GS_ADM_ILMOBJECTS
68	DBA_ILMPARAMETERS	GS_ADM_ILMPARAMETERS
69	DBA_ILMPOLICIES	GS_ADM_ILMPOLICIES
70	DBA_ILMRESULTS	GS_ADM_ILMRESULTS
71	DBA_ILMTASKS	GS_ADM_ILMTASKS
72	DBA_IND_COLUMNS	ADM_IND_COLUMNS

No.	Oracle Database	GaussDB
73	DBA_IND_EXPRESSIONS	ADM_IND_EXPRESSIONS
74	DBA_IND_PARTITIONS	ADM_IND_PARTITIONS
75	DBA_INDEXES	ADM_INDEXES
76	DBA_OBJECTS	ADM_OBJECTS
77	DBA_PART_INDEXES	ADM_PART_INDEXES
78	DBA_PROCEDURES	ADM_PROCEDURES
79	DBA_SCHEDULER_JOBS	ADM_SCHEDULER_JOBS
80	DBA_SCHEDULER_RUNNING_JOBS	ADM_SCHEDULER_RUNNING_JOBS
81	DBA_SEGMENTS	ADM_SEGMENTS
82	DBA_SEQUENCES	ADM_SEQUENCES
83	DBA_SOURCE	ADM_SOURCE
84	DBA_IND_SUBPARTITIONS	ADM_IND_SUBPARTITIONS
85	DBA_SUBPART_COL_STATISTICS	ADM_SUBPART_COL_STATISTICS
86	DBA_SUBPART_KEY_COLUMNS	ADM_SUBPART_KEY_COLUMNS
87	DBA_SYS_PRIVS	ADM_SYS_PRIVS
88	DBA_TAB_COL_STATISTICS	ADM_TAB_COL_STATISTICS
89	DBA_TAB_HISTOGRAMS	ADM_TAB_HISTOGRAMS
90	DBA_TAB_STATISTICS	ADM_TAB_STATISTICS
91	DBA_TAB_STATS_HISTORY	ADM_TAB_STATS_HISTORY
92	DBA_TABLESPACES	ADM_TABLESPACES
93	DBA_TYPES	ADM_TYPES
94	DBA_USERS	ADM_USERS
95	DBA_SYNONYMS	ADM_SYNONYMS
96	DBA_TAB_COLS	ADM_TAB_COLS
97	DBA_TAB_COLUMNS	ADM_TAB_COLUMNS
98	DBA_TAB_COMMENTS	ADM_TAB_COMMENTS
99	DBA_TABLES	ADM_TABLES
100	DBA_TAB_PARTITIONS	ADM_TAB_PARTITIONS

No.	Oracle Database	GaussDB
101	DBA_TAB_SUBPARTITIONS	ADM_TAB_SUBPARTITIONS
102	DBA_TRIGGERS	ADM_TRIGGERS
103	DBA_TYPE_ATTRS	ADM_TYPE_ATTRS
104	DBA_VIEWS	ADM_VIEWS
105	ROLE_ROLE_PRIVS	ROLE_ROLE_PRIVS
106	ROLE_SYS_PRIVS	ROLE_SYS_PRIVS
107	ROLE_TAB_PRIVS	ROLE_TAB_PRIVS
108	USER_COL_COMMENTS	MY_COL_COMMENTS
109	USER_COL_PRIVS	MY_COL_PRIVS
110	USER_COLL_TYPES	MY_COLL_TYPES
111	USER_CONSTRAINTS	MY_CONSTRAINTS
112	USER_DEPENDENCIES	MY_DEPENDENCIES
113	DICT	DICT
114	DICTIONARY	DICTIONARY
115	DUAL	DUAL
116	NLS_DATABASE_PARAMETERS	NLS_DATABASE_PARAMETERS
117	NLS_INSTANCE_PARAMETERS	NLS_INSTANCE_PARAMETERS
118	PLAN_TABLE	PLAN_TABLE
119	USER_ERRORS	MY_ERRORS
120	USER_ILMDATAMOVEMENT-POLICIES	GS_MY_ILMDATAMOVEMENTPOLICIES
121	USER_ILMEVALUATIONDE-TAILS	GS_MY_ILMEVALUATIONDETAILS
122	USER_ILMOBJECTS	GS_MY_ILMOBJECTS
123	USER_ILMPOLICIES	GS_MY_ILMPOLICIES
124	USER_ILMRESULTS	GS_MY_ILMRESULTS
125	USER_ILMTASKS	GS_MY_ILMTASKS
126	USER_IND_COLUMNS	MY_IND_COLUMNS
127	USER_IND_EXPRESSIONS	MY_IND_EXPRESSIONS
128	USER_IND_PARTITIONS	MY_IND_PARTITIONS

No.	Oracle Database	GaussDB
129	USER_IND_SUBPARTITIONS	MY_IND_SUBPARTITIONS
130	USER_INDEXES	MY_INDEXES
131	USER_JOBS	MY_JOBS
132	USER_OBJECTS	MY_OBJECTS
133	USER_PART_COL_STATISTICS	MY_PART_COL_STATISTICS
134	USER_PART_INDEXES	MY_PART_INDEXES
135	USER_PART_TABLES	MY_PART_TABLES
136	USER_PROCEDURES	MY_PROCEDURES
137	USER_RECYCLEBIN	MY_RECYCLEBIN
138	USER_SCHEDULER_JOB_ARGS	MY_SCHEDULER_JOB_ARGS
139	USER_SCHEDULER_PROGRAM_ARGS	MY_SCHEDULER_PROGRAM_ARGS
140	USER_SEQUENCES	MY_SEQUENCES
141	USER_SOURCE	MY_SOURCE
142	USER_SUBPART_KEY_COLUMNS	MY_SUBPART_KEY_COLUMNS
143	USER_SYNONYMS	MY_SYNONYMS
144	USER_SYS_PRIVS	MY_SYS_PRIVS
145	USER_TAB_COL_STATISTICS	MY_TAB_COL_STATISTICS
146	USER_TAB_COLUMNS	MY_TAB_COLUMNS
147	USER_TAB_COMMENTS	MY_TAB_COMMENTS
148	USER_TAB_HISTOGRAMS	MY_TAB_HISTOGRAMS
149	USER_TAB_PARTITIONS	MY_TAB_PARTITIONS
150	USER_TAB_STATISTICS	MY_TAB_STATISTICS
151	USER_TAB_STATS_HISTORY	MY_TAB_STATS_HISTORY
152	USER_TAB_SUBPARTITIONS	MY_TAB_SUBPARTITIONS
153	USER_TABLES	MY_TABLES
154	USER_TABLESPACES	MY_TABLESPACES
155	USER_TRIGGERS	MY_TRIGGERS
156	USER_TYPE_ATTRS	MY_TYPE_ATTRS
157	USER_TYPES	MY_TYPES

No.	Oracle Database	GaussDB
158	USER_VIEWS	MY_VIEWS
159	V$GLOBAL_TRANSACTION	V$GLOBAL_TRANSACTION
160	V$NLS_PARAMETERS	V$NLS_PARAMETERS
161	V$SESSION_WAIT	V$SESSION_WAIT
162	V$SYSSTAT	V$SYSSTAT
163	V$SYSTEM_EVENT	V$SYSTEM_EVENT
164	V$VERSION	V$VERSION
165	V$INSTANCE	V_INSTANCE
166	GV$INSTANCE	GV_INSTANCE
167	V$MYSTAT	V_MYSTAT
168	V$SESSION	V_SESSION
169	GV$SESSION	GV_SESSION
170	V$SESSION_LONGOPS	DV_SESSION_LONGOPS
171	V$SESSION	DV_SESSIONS
172	ALL_ARGUMENTS	DB_ARGUMENTS
173	USER_CONS_COLUMNS	MY_CONS_COLUMNS
174	USER_PART_KEY_COLUMNS	MY_PART_KEY_COLUMNS
175	USER_SUBPART_COL_STATISTICS	MY_SUBPART_COL_STATISTICS
176	USER_ROLE_PRIVS	MY_ROLE_PRIVS
177	DBA_TAB_PRIVS	ADM_TAB_PRIVS
178	USER_SCHEDULER_JOBS	MY_SCHEDULER_JOBS
179	V$LOCK	V$LOCK
180	V$DBLINK	V$DBLINK
181	V$OPEN_CURSOR	V$OPEN_CURSOR
182	ALL_TAB_PRIVS	DB_TAB_PRIVS
183	ALL_TAB_MODIFICATIONS	DB_TAB_MODIFICATIONS
184	USER_TAB_MODIFICATIONS	MY_TAB_MODIFICATIONS
185	USER_AUDIT_TRAIL	MY_AUDIT_TRAIL

13 Advanced Packages

Table 13-1 lists the advanced packages compatible with GaussDB.

Table 13-1 Supported advanced packages

No.	Oracle Database	GaussDB	Difference
1	DBMS_LOB	DBE_LOB	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_LOB" in *Developer Guide*.
2	DBMS_RANDOM	DBE_RANDOM	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_RANDOM" in *Developer Guide*.
3	DBMS_OUTPUT	DBE_OUTPUT	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_OUTPUT" in *Developer Guide*.
4	UTL_RAW	DBE_RAW	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_RAW" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
5	DBMS_SCHEDULER	DBE_SCHEDULER	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_SCHEDULER" in *Developer Guide*.
6	DBMS_UTILITY	DBE_UTILITY	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_UTILITY" in *Developer Guide*.
7	DBMS_SQL	DBE_SQL	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_SQL" in *Developer Guide*.
8	UTL_FILE	DBE_FILE	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_FILE" in *Developer Guide*.
9	DBMS_SESSION	DBE_SESSION	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_SESSION" in *Developer Guide*.
10	UTL_MATCH	DBE_MATCH	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_MATCH" in *Developer Guide*.
11	DBMS_APPLICATION_INFO	DBE_APPLICATION_INFO	For details about how to use it in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_APPLICATION_INFO" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
12	DBMS_XMLDOM	DBE_XMLDOM	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_XMLDOM" in *Developer Guide*.
13	DBMS_XMLPARSER	DBE_XMLPARSER	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_XMLPARSER" in *Developer Guide*.
14	DBMS_ILM	DBE_ILM	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_ILM" in *Developer Guide*.
15	DBMS_ILM_ADMIN	DBE_ILM_ADMIN	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_ILM_ADMIN" in *Developer Guide*.
16	DBMS_COMPRESSION	DBE_COMPRESSION	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_COMPRESSION" in *Developer Guide*.
17	DBMS_HEAT_MAP	DBE_HEAT_MAP	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_HEAT_MAP" in *Developer Guide*.
18	DBMS_DESCRIBE	DBE_DESCRIBE	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_DESCRIBE" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
19	DBMS_XMLGEN	DBE_XMLGEN	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_XMLGEN" in *Developer Guide*.
20	DBMS_STATS	DBE_STATS	For details about information in GaussDB, see "Stored Procedure > Advanced Packages > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*.

Table 13-2 DBMS_LOB compatibility

No.	Oracle Database	GaussDB	Difference
1	APPEND Procedures	APPEND Procedures	-
2	CLOB2FILE Procedure	Not supported.	-
3	CLOSE Procedure	BFILECLOSE Procedure	GaussDB: The parameter type is BFILE, and no function overloading exists. Oracle Database: There are three overloaded procedures. The procedures have three parameters **lob_loc**, **lob_loc**, and **file_loc**, which are of the BLOB, CLOB CHARACTER SET ANY_CS, and BFILE types, respectively.
4	COMPARE Functions	COMPARE Functions	GaussDB: There are three overloaded functions. The third parameter (**len**) is of the BIGINT type. Oracle Database: There are three overloaded functions. The third parameter (**amount**) is of the INTEGER type.

No.	Oracle Database	GaussDB	Difference
5	CONVERTTOBLOB Procedure	LOB_CONVERTTOBLOB Procedure	GaussDB: This procedure has five parameters, and the third, fourth, and fifth parameters are of the BIGINT type. Oracle Database: This procedure has eight parameters. In addition to all GaussDB parameters, the **blob_csid**, **lang_context**, and **warning** parameters are added, which are of the NUMBER, INTEGER, and INTEGER types, respectively. The third, fourth, and fifth parameters are of the INTEGER type.
6	CONVERTTOCLOB Procedure	LOB_CONVERTTOCLOB Procedure	GaussDB: This procedure has five parameters. The third, fourth, and fifth parameters are of the BIGINT type. Oracle Database: This procedure has eight parameters. The third, fourth, and fifth parameters are of the INTEGER type. In addition to all GaussDB parameters, this procedure in Oracle Database adds three parameters: **blob_csid**, **lang_context**, and **warning**, which are of NUMBER, INTEGER, and INTEGER types, respectively.
7	COPY Procedures	LOB_COPY Functions	-
8	COPY_DBFS_LINK Procedures	Not supported.	-
9	COPY_FROM_DBFS_LINK	Not supported.	-

No.	Oracle Database	GaussDB	Difference
10	CREATETEMPORARY Procedures	CREATE_TEMPORARY Procedures	GaussDB: There are two overloaded procedures. The first parameter (**lob_loc**) of the first overloaded procedure is of the BLOB type, and the first parameter (**lob_loc**) of the second overloaded procedure is of the CLOB type. The third parameter (**dur**) of the two overloaded procedures is of the INTEGER type, and the default value is **10**. Oracle Database: There are two overloaded procedures. The first parameter (**lob_loc**) of the first overloaded procedure is of the BLOB type. The first parameter (**lob_loc**) of the second overloaded procedure is of the CLOB type. The third parameter (**dur**) of the two overloaded procedures is of the PLS_INTEGER type. The default value of **dur** of the first overloaded procedure is **DBMS_LOB.SESSION**, and the default value of **dur** of the second overloaded procedure is **10**.
11	DBFS_LINK_GENERATE_PATH Functions	Not supported.	-
12	ERASE Procedures	LOB_ERASE Procedures	-
13	FILECLOSE Procedure	Not supported.	-
14	FILECLOSEALL Procedure	Not supported.	-
15	FILEEXISTS Function	Not supported.	-
16	FILEGETNAME Procedure	Not supported.	-
17	FILEISOPEN Function	Not supported.	-
18	FILEOPEN Procedure	Not supported.	-

No.	Oracle Database	GaussDB	Difference
19	FRAGMENT_DELETE Procedure	Not supported.	-
20	FRAGMENT_INSERT Procedures	Not supported.	-
21	FRAGMENT_MOVE Procedure	Not supported.	-
22	FRAGMENT_REPLACE Procedures	Not supported.	-
23	FREETEMPORARY Procedures	Not supported.	-
24	GET_DBFS_LINK Functions	Not supported.	-
25	GET_DBFS_LINK_STATE Procedures	Not supported.	-
26	GETCHUNKSIZE Functions	GETCHUNKSIZE Functions	-
27	GETCONTENTTYPE Functions	Not supported.	-
28	GETLENGTH Functions	Not supported.	-
29	GETOPTIONS Functions	Not supported.	-
30	GET_STORAGE_LIMIT Function	Not supported.	-
31	INSTR Functions	MATCH Functions	GaussDB: There are three overloaded functions. The third and fourth parameters of the three overloaded functions are of the BIGINT type. Oracle Database: There are three overloaded functions. The third and fourth parameters of the three overloaded functions are of the INTEGER type.

No.	Oracle Database	GaussDB	Difference
32	ISOPEN Functions	Not supported.	-
33	ISREMOTE Function	Not supported.	-
34	ISSECUREFILE Function	Not supported.	-
35	ISTEMPORARY Functions	Not supported.	-
36	LOADBLOBFROMFILE Procedure	LOADBLOBFROMBFILE Procedure	-
37	LOADCLOBFROMFILE Procedure	LOADCLOBFROMBFILE Procedure	-
38	LOADFROMFILE Procedure	LOADFROMBFILE Procedure	-
39	MOVE_TO_DBFS_LINK Procedures	Not supported.	-
40	OPEN Procedures	BFILEOPEN Procedure	GaussDB: There is no overloaded procedure. The first parameter (**bfile**) is of the DBE_LOB.BFILE type, and the second parameter (**open_mode**) is of the TEXT type. Only the read mode is supported. Oracle Database: There are three overloaded procedures. In the first overloaded procedure, the first parameter (**lob_loc**) is of the NOCOPY BLOB type, and the second parameter (**openmode**) is of the BINARY_INTEGER type. In the second overloaded procedure, the first parameter (**lob_loc**) is of the NOCOPY CLOB CHARACTER SET ANY_CS type, and the second parameter (**openmode**) is of the BINARY_INTEGER type. In the third overloaded procedure, the first parameter (**file_loc**) is of the NOCOPY BFILE type, and the second parameter (**openmode**) is of the BINARY_INTEGER type, and the value can only be **file_readonly**.

No.	Oracle Database	GaussDB	Difference
41	READ Procedures	READ Procedures	GaussDB: There are two overloaded procedures. Oracle Database: There are three overloaded procedures. The first two overloaded procedures are the same as those in GaussDB. The third overloaded procedure includes four parameters: **file_loc**, **amount**, **offset**, and **buffer**, which are of the BFILE, NOCOPY INTEGER, INTEGER, and RAW types, respectively.
42	SET_DBFS_LINK Procedures	Not supported.	-
43	SETCONTENTTYPE Procedure	Not supported.	-
44	SETOPTIONS Procedures	Not supported.	-
45	SUBSTR Functions	LOB_SUBSTR Functions	-
46	TRIM Procedures	STRIP Functions	GaussDB: There are two overloaded procedures. The second parameter (**newlen**) of the two overloaded procedures is of the BIGINT type. Oracle Database: There are two overloaded procedures. The second parameter (**newlen**) of the two overloaded procedures is of the INTEGER type.
47	WRITE Procedures	WRITE Functions	-
48	WRITEAPPEND Procedures	WRITEAPPEND Functions	-

Table 13-3 DBMS_RANDOM compatibility

No.	Oracle Database	GaussDB	Difference
1	INITIALIZE Procedure	Not supported.	-

No.	Oracle Database	GaussDB	Difference
2	NORMAL Function	Not supported.	-
3	RANDOM Function	Not supported.	-
4	SEED Procedures	DBE_RANDOM.SET_SEED Function	GaussDB: There is no overloaded function. The parameter is of the INTEGER type. Oracle Database: There are two overloaded procedures. The parameter types of the two overloaded procedures are VARCHAR2 and BINARY_INTEGER, respectively.
5	STRING Function	Not supported.	-
6	TERMINATE Procedure	Not supported.	-
7	VALUE Functions	DBE_RANDOM.GET_VALUE Function	GaussDB: There is no overloaded function. Oracle Database: The VALUE function without parameters is overloaded, and the return type is NUMBER.

Table 13-4 DBMS_OUTPUT compatibility

No.	Oracle Database	GaussDB	Difference
1	DISABLE Procedure	DISABLE Function	-
2	ENABLE Procedure	ENABLE Function	-
3	GET_LINE Procedure	GET_LINE Function	-

No.	Oracle Database	GaussDB	Difference
4	GET_LINES Procedure	GET_LINES Function	GaussDB: There is no overloaded function. The first parameter (**lines**) is of the VARCHAR[] type. Oracle Database: There are two overloaded procedures. The first parameters (**lines**) of the two overloaded procedures are of the CHARARR and DBMSOUTPUT_LINESARRAY types, respectively.
5	NEW_LINE Procedure	NEW_LINE Function	-
6	PUT Procedure	PUT Function	GaussDB: If the character set of the database server (**server_encoding**) is not encoded in UTF-8 and the character encoding of the input parameter is valid UTF-8, this function converts character encoding based on the relationship "UTF8 > server_encoding" and then outputs the result regardless of the data type of the input parameter. Oracle Database: If the character set of the database server (**server_encoding**) is not encoded in UTF-8, the character encoding of the input parameter is valid UTF-8, and the input parameter type is NVARCHAR2, this procedure coverts the character encoding based on the relationship "UTF8 > server_encoding" and then outputs the result. If the input parameter is of other character types, the character encoding is regarded as an invalid character and output as a placeholder.

No.	Oracle Database	GaussDB	Difference
7	PUT_LINE Procedure	PUT_LINE Function	GaussDB: If the character set of the database server (**server_encoding**) is not encoded in UTF-8 and the character encoding of the input parameter is valid UTF-8, this function converts character encoding based on the relationship "UTF8 > server_encoding" and then outputs the result regardless of the data type of the input parameter. Oracle Database: If the character set of the database server (**server_encoding**) is not encoded in UTF-8, the character encoding of the input parameter is valid UTF-8, and the input parameter type is NVARCHAR2, this procedure coverts the character encoding based on the relationship "UTF8 > server_encoding" and then outputs the result. If the input parameter is of other character types, the character encoding is regarded as an invalid character and output as a placeholder.

Table 13-5 UTL_RAW compatibility

No.	Oracle Database	GaussDB	Difference
1	BIT_AND Function	BIT_AND Function	-
2	BIT_COMPLEMENT Function	BIT_COMPLEMENT Function	-
3	BIT_OR Function	BIT_OR Function	GaussDB: The two parameters are defined as the TEXT type and returned as the TEXT type. Oracle Database: The two parameters are of the RAW type and returned as the RAW type.
4	BIT_XOR Function	BIT_XOR Function	-
5	CAST_FROM_BINARY_DOUBLE Function	CAST_FROM_BINARY_DOUBLE_TO_RAW Function	-

No.	Oracle Database	GaussDB	Difference
6	CAST_FROM_BINARY_FLOAT Function	CAST_FROM_BINARY_FLOAT_TO_RAW Function	GaussDB: The **n** parameter is of the FLOAT4 type. Oracle Database: The **n** parameter is of the FLOAT type.
7	CAST_FROM_BINARY_INTEGER Function	CAST_FROM_BINARY_INTEGER_TO_RAW Function	GaussDB: The **value** parameter is of the BIGINT type. Oracle Database: The **value** parameter is of the INTEGER type.
8	CAST_FROM_NUMBER Function	CAST_FROM_NUMBER_TO_RAW Function	GaussDB: The **n** parameter is of the NUMERIC type. Oracle Database: The **n** parameter is of the NUMBER type.
9	CAST_TO_BINARY_DOUBLE Function	CAST_FROM_RAW_TO_BINARY_DOUBLE Function	-
10	CAST_TO_BINARY_FLOAT Function	CAST_FROM_RAW_TO_BINARY_FLOAT Function	GaussDB: The function returns the FLOAT4 type. Oracle Database: The function returns the FLOAT type.
11	CAST_TO_BINARY_INTEGER Function	CAST_FROM_RAW_TO_BINARY_INTEGER Function	GaussDB: The **endianess** parameter is of the INTEGER type, and the function returns the INTEGER type. Oracle Database: The **endianess** parameter is of the PLS_INTEGER type, and the function returns the BINARY_INTEGER type.
12	CAST_TO_NUMBER Function	CAST_FROM_RAW_TO_NUMBER Function	GaussDB: The function returns the NUMERIC type. Oracle Database: The function returns the NUMBER type.
13	CAST_TO_NVARCHAR2 Function	CAST_FROM_RAW_TO_NVARCHAR2 Function	-

No.	Oracle Database	GaussDB	Difference
14	CAST_TO_RAW Function	CAST_FROM_VARCHAR2_TO_RAW Function	-
15	CAST_TO_VARCHAR2 Function	CAST_TO_VARCHAR2 Function	-
16	COMPARE Function	COMPARE Function	GaussDB: The function returns the INTEGER type. Oracle Database: The function returns the NUMBER type.
17	CONCAT Function	CONCAT Function	-
18	CONVERT Function	CONVERT Function	-
19	COPIES Function	COPIES Function	GaussDB: The **n** parameter is of the NUMERIC type. Oracle Database: The **n** parameter is of the NUMBER type.
20	LENGTH Function	GET_LENGTH Function	GaussDB: The function returns the INTEGER type. Oracle Database: The function returns the NUMBER type.
21	OVERLAY Function	OVERLAY Function	-
22	REVERSE Function	REVERSE Function	-
23	SUBSTR Function	SUBSTR Function	GaussDB: The **lob_loc** parameter is of the BLOB type. The **off_set** parameter is of the INTEGER type and its default value is **1**. The **amount** parameter is of the INTEGER type and its default value is **32767**. Oracle Database: The **r** parameter is of the RAW type. The **pos** parameter is of the BINARY_INTEGER type and has no default value. The **len** parameter is of the BINARY_INTEGER type and its default value is **NULL**.

No.	Oracle Database	GaussDB	Difference
24	TRANSLATE Function	TRANSLATE Function	-
25	TRANSLITERATE Function	TRANSLITERATE Function	-
26	XRANGE Function	XRANGE Function	GaussDB: The **start_byte** and **end_byte** parameters do not have default values. Oracle Database: The default values of the **start_byte** and **end_byte** parameters are **NULL**.

Table 13-6 DBMS_SCHEDULER compatibility

No.	Oracle Database	GaussDB
1	ADD_EVENT_QUEUE_SUBSCRIBER Procedure	Not supported.
2	ADD_GROUP_MEMBER Procedure	Not supported.
3	ADD_JOB_EMAIL_NOTIFICATION Procedure	Not supported.
4	ADD_TO_INCOMPATIBILITY Procedure	Not supported.
5	ALTER_CHAIN Procedure	Not supported.
6	ALTER_RUNNING_CHAIN Procedure	Not supported.
7	CLOSE_WINDOW Procedure	Not supported.
8	COPY_JOB Procedure	Not supported.
9	CREATE_CHAIN Procedure	Not supported.
10	CREATE_CREDENTIAL Procedure	CREATE_CREDENTIAL Procedure
11	CREATE_DATABASE_DESTINATION Procedure	Not supported.
12	CREATE_EVENT_SCHEDULE Procedure	Not supported.
13	CREATE_FILE_WATCHER Procedure	Not supported.

No.	Oracle Database	GaussDB
14	CREATE_GROUP Procedure	Not supported.
15	CREATE_INCOMPATIBILITY Procedure	Not supported.
16	CREATE_JOB Procedure	CREATE_JOB Procedure
17	CREATE_JOB_CLASS Procedure	CREATE_JOB_CLASS Procedure
18	CREATE_JOBS Procedure	Not supported.
19	CREATE_PROGRAM Procedure	CREATE_PROGRAM Procedure
20	CREATE_RESOURCE Procedure	Not supported.
21	CREATE_SCHEDULE Procedure	CREATE_SCHEDULE Procedure
22	CREATE_WINDOW Procedure	Not supported.
23	DEFINE_ANYDATA_ARGUMENT Procedure	Not supported.
24	DEFINE_CHAIN_EVENT_STEP Procedure	Not supported.
25	DEFINE_CHAIN_RULE Procedure	Not supported.
26	DEFINE_CHAIN_STEP Procedure	Not supported.
27	DEFINE_METADATA_ARGUMENT Procedure	Not supported.
28	DEFINE_PROGRAM_ARGUMENT Procedure	DEFINE_PROGRAM_ARGUMENT Procedure
29	DISABLE Procedure	DISABLE Procedure
30	DROP_AGENT_DESTINATION Procedure	Not supported.
31	DROP_CHAIN Procedure	Not supported.
32	DROP_CHAIN_RULE Procedure	Not supported.
33	DROP_CHAIN_STEP Procedure	Not supported.
34	DROP_CREDENTIAL Procedure	DROP_CREDENTIAL Procedure
35	DROP_DATABASE_DESTINATION Procedure	Not supported.
36	DROP_FILE_WATCHER Procedure	Not supported.
37	DROP_GROUP Procedure	Not supported.

No.	Oracle Database	GaussDB
38	DROP_INCOMPATIBILITY Procedure	Not supported.
39	DROP_JOB Procedure	DROP_JOB Procedure
40	DROP_JOB_CLASS Procedure	DROP_JOB_CLASS Procedure
41	DROP_PROGRAM Procedure	DROP_PROGRAM Procedure
42	DROP_PROGRAM_ARGUMENT Procedure	Not supported.
43	DROP_SCHEDULE Procedure	DROP_SCHEDULE Procedure
44	DROP_WINDOW Procedure	Not supported.
45	ENABLE Procedure	ENABLE Procedure
46	END_DETACHED_JOB_RUN Procedure	Not supported.
47	EVALUATE_CALENDAR_STRING Procedure	EVALUATE_CALENDAR_STRING Procedure
48	EVALUATE_RUNNING_CHAIN Procedure	Not supported.
49	GENERATE_JOB_NAME Function	GENERATE_JOB_NAME Function
50	GET_AGENT_INFO Function	Not supported.
51	GET_AGENT_VERSION Function	Not supported.
52	GET_ATTRIBUTE Procedure	Not supported.
53	GET_FILE Procedure	Not supported.
54	GET_SCHEDULER_ATTRIBUTE Procedure	Not supported.
55	OPEN_WINDOW Procedure	Not supported.
56	PURGE_LOG Procedure	Not supported.
57	PUT_FILE Procedure	Not supported.
58	REMOVE_EVENT_QUEUE_SUBSCRIBER Procedure	Not supported.
59	REMOVE_FROM_INCOMPATIBILITY Procedure	Not supported.
60	REMOVE_GROUP_MEMBER Procedure	Not supported.

No.	Oracle Database	GaussDB
61	REMOVE_JOB_EMAIL_NOTIFICATION Procedure	Not supported.
62	RESET_JOB_ARGUMENT_VALUE Procedure	Not supported.
63	RUN_CHAIN Procedure	Not supported.
64	RUN_JOB Procedure	RUN_JOB Procedure
65	SET_AGENT_REGISTRATION_PASS Procedure	Not supported.
66	SET_ATTRIBUTE Procedure	SET_ATTRIBUTE Procedure
67	SET_ATTRIBUTE_NULL Procedure	Not supported.
68	SET_JOB_ANYDATA_VALUE Procedure	Not supported.
69	SET_JOB_ARGUMENT_VALUE Procedure	SET_JOB_ARGUMENT_VALUE Procedure
70	SET_JOB_ATTRIBUTES Procedure	Not supported.
71	SET_RESOURCE_CONSTRAINT Procedure	Not supported.
72	SET_SCHEDULER_ATTRIBUTE Procedure	Not supported.
73	STOP_JOB Procedure	STOP_JOB Procedure

Table 13-7 DBMS_UTILITY compatibility

No.	Oracle Database	GaussDB	Difference
1	ACTIVE_INSTANCES Procedure	Not supported.	-
2	ANALYZE_DATABASE Procedure	Not supported.	-
3	ANALYZE_PART_OBJECT Procedure	Not supported.	-
4	ANALYZE_SCHEMA Procedure	Not supported.	-

No.	Oracle Database	GaussDB	Difference
5	CANONICALIZE Procedure	CANONICALIZE Procedure	GaussDB: The default size of the **canon_len** parameter is 1024 bytes. Oracle Database: The **canon_len** parameter has no default value.
6	COMMA_TO_TABLE Procedures	COMMA_TO_TABLE Procedure	GaussDB: The **tab** parameter is a VARCHAR2 array. Oracle Database: There are two overloaded procedures. The **tab** parameter can be of the uncl_array or lname_array type.
7	COMPILE_SCHEMA Procedure	Not supported.	-
8	CREATE_ALTER_TYPE_ERROR_TABLE Procedure	Not supported.	-
9	CURRENT_INSTANCE Function	Not supported.	-
10	DATA_BLOCK_ADDRESS_BLOCK Function	Not supported.	-
11	DATA_BLOCK_ADDRESS_FILE Function	Not supported.	-
12	DB_VERSION Procedure	DB_VERSION Procedure	GaussDB: There is only the **version** parameter, which is of the VARCHAR2 type. Oracle Database: There are the **version** and **compatibility** parameters, which are of the VARCHAR2 type.
13	EXEC_DDL_STATEMENT Procedure	EXEC_DDL_STATEMENT Function	GaussDB: The **parse_string** parameter is of the TEXT type. Oracle Database: The **parse_string** parameter is of the VARCHAR2 type.

No.	Oracle Database	GaussDB	Difference
14	EXPAND_SQL_TEXT Procedure	EXPAND_SQL_TEXT Function	GaussDB: The **output_sql_text** parameter is of the CLOB type. Oracle Database: The **output_sql_text** parameter is of the NOCOPY CLOB type. The **OUT** parameter is transferred by reference.
15	FORMAT_CALL_STACK Function	FORMAT_CALL_STACK Function	GaussDB: The function returns the TEXT type. Oracle Database: The function returns the VARCHAR2 type.
16	FORMAT_ERROR_BACKTRACE Function	FORMAT_ERROR_BACKTRACE Function	GaussDB: The function returns the TEXT type. Oracle Database: The function returns the VARCHAR2 type.
17	FORMAT_ERROR_STACK Function	FORMAT_ERROR_STACK Function	GaussDB: The function returns the TEXT type. Oracle Database: The function returns the VARCHAR2 type.
18	GET_CPU_TIME Function	GET_CPU_TIME Function	GaussDB: The function returns the BIGINT type. Oracle Database: The function returns the NUMBER type.
19	GET_DEPENDENCY Procedure	Not supported.	-
20	GET_ENDIANNESS Function	GET_ENDIANNESS Function	GaussDB: The function returns the INTEGER type. Oracle Database: The function returns the NUMBER type.
21	GET_HASH_VALUE Function	GET_HASH_VALUE Function	GaussDB: The **base** and **hash_size** parameters and the return types are all INTEGER. Oracle Database: The **base** and **hash_size** parameters and the return types are all NUMBER.

No.	Oracle Database	GaussDB	Difference
22	GET_PARAMETER_VALUE Function	Not supported.	-
23	GET_SQL_HASH Function	GET_SQL_HASH Function	GaussDB: The **last4bytes** parameter of the BIGINT type specifies the last four bytes of an MD5 hash value and is displayed as an unsigned integer. The function returns the BIGINT type. Oracle Database: The **pre10ihash** parameter of the NUMBER type is used to store the 4-byte hash value among the 16 bytes calculated by MD5.
24	GET_TIME Function	GET_TIME Function	GaussDB: The function returns the BIGINT type. Oracle Database: The function returns the NUMBER type.
25	GET_TZ_TRANSITIONS Procedure	Not supported.	-
26	INVALIDATE Procedure	Not supported.	-
27	IS_BIT_SET Function	IS_BIT_SET Function	GaussDB: The **n** parameter and return value type are both INTEGER. Oracle Database: The **n** parameter and return value type are both NUMBER.
28	IS_CLUSTER_DATABASE Function	IS_CLUSTER_DATABASE Function	-
29	MAKE_DATA_BLOCK_ADDRESS Function	Not supported.	-

No.	Oracle Database	GaussDB	Difference
30	NAME_RESOLVE Procedure	NAME_RESOLVE Procedure	GaussDB: The **context** and **part1_type** parameters are of the INTEGER type, and the **object_number** parameter is of the OID type. GaussDB does not support implicit conversion from NUMBER to OID. Oracle Database: The **context**, **part1_type**, and **object_number** parameters are of the NUMBER type.
31	NAME_TOKENIZE Procedure	NAME_TOKENIZE Procedure	GaussDB: The **nextpos** parameter is of the INTEGER type. Oracle Database: The **nextpos** parameter is of the BINARY_INTEGER type.
32	OLD_CURRENT_SCHEMA Function	OLD_CURRENT_SCHEMA Function	GaussDB: The function returns the VARCHAR type. Oracle Database: The function returns the VARCHAR2 type.
33	OLD_CURRENT_USER Function	OLD_CURRENT_USER Function	GaussDB: The function returns the TEXT type. Oracle Database: The function returns the VARCHAR2 type.
34	PORT_STRING Function	Not supported.	-
35	SQLID_TO_SQLHASH Function	Not supported.	-
36	TABLE_TO_COMMA Procedures	TABLE_TO_COMMA Procedure	GaussDB: The **tab** parameter is a VARCHAR2 array. Oracle Database: There are two overloaded stored procedures. The **tab** parameter can be of the uncl_array or lname_array type.
37	VALIDATE Procedure	Not supported.	-

No.	Oracle Database	GaussDB	Difference
38	WAIT_ON_PENDING_DML Function	Not supported.	-

Table 13-8 DBMS_SQL compatibility

No.	Oracle Database	GaussDB	Difference
1	BIND_ARRAY Procedures	SQL_BIND_ARRAY Function	-
2	BIND_VARIABLE Procedures	SQL_BIND_VARIABLEFunction	-
3	BIND_VARIABLE_PKG Procedure	Not supported.	-
4	CLOSE_CURSOR Procedure	SQL_UNREGISTER_CONTEXT Function	-
5	COLUMN_VALUE Procedure	GET_RESULT Procedure	-
6	COLUMN_VALUE_LONG Procedure	Not supported.	-
7	DEFINE_ARRAY Procedure	SET_RESULTS_TYPE Procedure	-
8	DEFINE_COLUMN Procedures	SET_RESULT_TYPE Procedure	-
9	DEFINE_COLUMN_CHAR Procedure	Not supported.	-
10	DEFINE_COLUMN_LONG Procedure	Not supported.	-
11	DEFINE_COLUMN_RAW Procedure	Not supported.	-
12	DEFINE_COLUMN_ROWID Procedure	Not supported.	-
13	DESCRIBE_COLUMNS Procedure	DESCRIBE_COLUMNS Procedure	-
14	DESCRIBE_COLUMNS2 Procedure	Not supported.	-
15	DESCRIBE_COLUMNS3 Procedure	Not supported.	-

No.	Oracle Database	GaussDB	Difference
16	EXECUTE Function	SQL_RUN Function	GaussDB: The return value is a constant **1**. Currently, the comparison between unknown types in the statement cannot return correct results. Oracle Database: The return value is the number of affected rows for INSERT, UPDATE, and DELETE statements and is meaningless for other statements.
17	EXECUTE_AND_FETCH Function	RUN_AND_NEXT Function	-
18	FETCH_ROWS Function	NEXT_ROW Function	-
19	GET_NEXT_RESULT Procedures	Not supported.	-
20	IS_OPEN Function	IS_ACTIVE Function	-
21	LAST_ERROR_POSITIO N Function	Not supported.	-
22	LAST_ROW_COUNT Function	LAST_ROW_COUNT Function	-
23	LAST_ROW_ID Function	Not supported.	-
24	LAST_SQL_FUNCTION_ CODE Function	Not supported.	-
25	OPEN_CURSOR Functions	REGISTER_CONTEXT Function	-
26	PARSE Procedures	Supported, with differences.	In GaussDB, the SQL_SET_SQL function does not support overloading.
27	RETURN_RESULT Procedures	Not supported.	-
28	TO_CURSOR_NUMBER Function	Not supported.	-
29	TO_REFCURSOR Function	Not supported.	-

No.	Oracle Database	GaussDB	Difference
30	VARIABLE_VALUE Procedures	GET_VARIABLE_RESULT Procedures	-
31	VARIABLE_VALUE_PKG Procedure	Not supported.	-

Table 13-9 DBMS_SQL data type compatibility

No.	Oracle Database	GaussDB
1	DBMS_SQL DESC_REC	DBE_SQL.DESC_REC
2	DBMS_SQL DATE_TABLE	DBE_SQL.DATE_TABLE
3	DBMS_SQL NUMBER_TABLE	DBE_SQL.NUMBER_TABLE
4	DBMS_SQL VARCHAR2_TABLE	DBE_SQL.VARCHAR2_TABLE
5	DBMS_SQL BLOB_TABLE	DBE_SQL.BLOB_TABLE

Table 13-10 UTL_FILE compatibility

No.	Oracle Database	GaussDB	Difference
1	FCLOSE Procedure	CLOSE Procedure	-
2	FCLOSE_ALL Procedure	CLOSE_ALL Procedure	-
3	FCOPY Procedure	COPY Procedure	-
4	FFLUSH Procedure	FLUSH Procedure	-
5	FGETATTR Procedure	GET_ATTR Procedure	-
6	FGETPOS Function	GET_POS Function	-
7	FOPEN Function	FOPEN Function	-
8	FOPEN_NCHAR Function	FOPEN_NCHAR Function	-
9	FREMOVE Procedure	REMOVE Procedure	-

No.	Oracle Database	GaussDB	Difference
10	FRENAME Procedure	RENAME Procedure	-
11	FSEEK Procedure	SEEK Procedure	-
12	GET_LINE Procedure	READ_LINE Procedure	-
13	GET_LINE_NCHAR Procedure	READ_LINE_NCHAR Procedure	-
14	GET_RAW Procedure	GET_RAW Procedure	-
15	IS_OPEN Function	IS_OPEN Function	-
16	NEW_LINE Procedure	Supported, with differences in the NEW_LINE function	GaussDB defines the API as a function.
17	PUT Procedure	Supported, with differences in the WRITE function	GaussDB defines the API as a function.
18	PUT_LINE Procedure	Supported, with differences in the WRITE_LINE function	GaussDB defines the API as a function.
19	PUT_LINE_NCHAR Procedure	Supported, with differences in the WRITE_LINE_NCHAR function	GaussDB defines the API as a function.
20	PUT_NCHAR Procedure	Supported, with differences in the WRITE_NCHAR function	GaussDB defines the API as a function.
21	PUTF Procedure	Supported, with differences in the FORMAT_WRITE function	GaussDB defines the API as a function.
22	PUTF_NCHAR Procedure	Supported, with differences in the FORMAT_WRITE_NCHAR function	GaussDB defines the API as a function.
23	PUT_RAW Procedure	Supported, with differences in the PUT_RAW function	GaussDB defines the API as a function.

Table 13-11 DBMS_SESSION compatibility

No.	Oracle Database	GaussDB	Difference
1	CLEAR_ALL_CONTEXT Procedure	Not supported.	-
2	CLEAR_CONTEXT Procedure	CLEAR_CONTEXT Function	-
3	CLEAR_IDENTIFIER Procedure	Not supported.	-
4	CLOSE_DATABASE_LINK Procedure	Not supported.	-
5	CURRENT_IS_ROLE_ENABLED Function	Not supported.	-
6	FREE_UNUSED_USER_MEMORY Procedure	Not supported.	-
7	GET_PACKAGE_MEMORY_UTILIZATION Procedure	Not supported.	-
8	IS_ROLE_ENABLED Function	Not supported.	-
9	IS_SESSION_ALIVE Function	Not supported.	-
10	LIST_CONTEXT Procedures	Not supported.	-
11	MODIFY_PACKAGE_STATE Procedure	MODIFY_PACKAGE_STATE Procedure	GaussDB: The scenario where **flags** is set to **1** is supported. Oracle Database: The scenario where **flags** is set to **1** or **2** is supported.
12	RESET_PACKAGE Procedure	Not supported.	-
13	SESSION_IS_ROLE_ENABLED Function	Not supported.	-
14	SESSION_TRACE_DISABLE Procedure	Not supported.	-
15	SESSION_TRACE_ENABLE Procedure	Not supported.	-

No.	Oracle Database	GaussDB	Difference
16	SET_CONTEXT Procedure	SET_CONTEXT Function	GaussDB: There are the **namespace**, **attribute**, and **value** parameters of the TEXT type. Oracle Database: There are the **namespace**, **attribute**, **value**, **username**, and **client_id** parameters of the VARCHAR2 type.
17	SET_EDITION_DEFERRED Procedure	Not supported.	-
18	SET_IDENTIFIER Procedure	Not supported.	-
19	SET_NLS Procedure	Not supported.	-
20	SET_ROLE Procedure	Not supported.	-
21	SET_SQL_TRACE Procedure	Not supported.	-
22	SLEEP Procedure	Not supported.	-
23	SWITCH_CURRENT_CONSUMER_GROUP Procedure	Not supported.	--
24	UNIQUE_SESSION_ID Function	Not supported.	-

Table 13-12 UTL_MATCH compatibility

No.	Oracle Database	GaussDB	Difference
1	EDIT_DISTANCE Function	Not supported.	-
2	EDIT_DISTANCE_SIMILARITY Function	EDIT_DISTANCE_SIMILARITY Function	GaussDB: The **str1** and **str2** parameters are of the TEXT type, and the function returns the INTEGER type. Oracle Database: The **s1** and **s2** parameters are of the VARCHAR2 type, and the function returns the PLS_INTEGER type.
3	JARO_WINKLER Function	Not supported.	-

No.	Oracle Database	GaussDB	Difference
4	JARO_WINKLER_SIMILARITY Function	Not supported.	-

Table 13-13 DBMS_APPLICATION_INFO compatibility

No.	Oracle Database	GaussDB	Difference
1	READ_CLIENT_INFO Procedure	READ_CLIENT_INFO Function	GaussDB: The **client_info** parameter is of the TEXT type. Oracle Database: The **client_info** parameter is of the VARCHAR2 type.
2	READ_MODULE Procedure	READ_MODULE Procedure	GaussDB: The **module_name** and **action_name** parameters are of the TEXT type. Oracle Database: The **module_name** and **action_name** parameters are of the VARCHAR2 type.
3	SET_ACTION Procedure	SET_ACTION Procedure	GaussDB: The **action_name** parameter is of the TEXT type. Oracle Database: The **action_name** parameter is of the VARCHAR2 type.
4	SET_CLIENT_INFO Procedure	SET_CLIENT_INFO Function	GaussDB: The **str** parameter is of the TEXT type, and the return type is VOID. Oracle Database: The **client_info** parameter is of the VARCHAR2 type and no value is returned. Both of them are written to the client. The maximum length is 64 bytes. If the length exceeds 64 bytes, it will be truncated.

No.	Oracle Database	GaussDB	Difference
5	SET_MODULE Procedure	SET_MODULE Procedure	GaussDB: The **module_name** and **action_name** parameters are of the TEXT type. Oracle Database: The **module_name** and **action_name** parameters are of the VARCHAR2 type.
6	SET_SESSION_LONGOPS Procedure	Not supported.	-

Table 13-14 DBMS_XMLDOM compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_XMLDOM.APPENDCHILD	DBE_XMLDOM.APPENDCHILD	• GaussDB: The error message "operation not support" is displayed for the APPEND ATTR node under the DOCUMENT node. Oracle Database: No error is reported in this scenario, but the mounting fails. • GaussDB: The error message "operation not support" is displayed for the APPEND ATTR node under the ATTR node. Oracle Database: No error is reported in this scenario, but the mounting fails. • GaussDB: When multiple child nodes of the ATTR type are added to a parent node, the child nodes with the same key value cannot exist under the same parent node. Oracle Database: Child nodes with the same key value can exist under the same parent node.

No.	Oracle Database	GaussDB	Difference
2	DBMS_XMLDOM.CREATEELEMENT	DBE_XMLDOM.CREATEELEMENT	-
3	DBMS_XMLDOM.CREATETEXTNODE	DBE_XMLDOM.CREATETEXTNODE	-
4	DBMS_XMLDOM.FREEDOCUMENT	DBE_XMLDOM.FREEDOCUMENT	GaussDB: Objects are not released immediately. They are released after a certain number of objects are accumulated. All nodes in the document are invalid. Oracle Database: The object is released immediately.
5	DBMS_XMLDOM.FREEELEMENT	DBE_XMLDOM.FREEEELEMENT	-
6	DBMS_XMLDOM.FREENODE	DBE_XMLDOM.FREENODE	-
7	DBMS_XMLDOM.FREENODELIST	DBE_XMLDOM.FREENODELIST	GaussDB: The nodelist will be released. Oracle Database: After the nodelist is released, it can still be queried in the original document.
8	DBMS_XMLDOM.GETATTRIBUTE	DBE_XMLDOM.GETATTRIBUTE	-
9	DBMS_XMLDOM.GETATTRIBUTES	DBE_XMLDOM.GETATTRIBUTES	-
10	DBMS_XMLDOM.GETCHILDNODES	DBE_XMLDOM.GETCHILDNODES	GaussDB: When the document node is used, DTD is included. Oracle Database: DTD is not included.
11	DBMS_XMLDOM.GETCHILDRENBYTAGNAME	DBE_XMLDOM.GETCHILDRENBYTAGNAME	GaussDB: The **ns** parameter of the DBE_XMLDOM.GETCHILDRENBYTAGNAME API does not support the asterisk (*) parameter. To obtain all attributes of a node, use the DBE_XMLDOM.GETCHILDNODES API. Oracle Database: The input parameter ***** is supported.

No.	Oracle Database	GaussDB	Difference
12	DBMS_XMLDOM.GETDOCUMENTELEMENT	DBE.XMLDOM.GETDOCUMENTELEMENT	-
13	DBMS_XMLDOM.GETFIRSTCHILD	DBE_XMLDOM.GETFIRSTCHILD	-
14	DBMS_XMLDOM.GETLASTCHILD	DBE_XMLDOM.GETLASTCHILD	-
15	DBMS_XMLDOM.GETLENGTH	DBE_XMLDOM.GETLENGTH	-
16	DBMS_XMLDOM.GETLOCALNAME	DBE_XMLDOM.GETLOCALNAME	-
17	DBMS_XMLDOM.GETNAMEDITEM	DBE_XMLDOM.GETNAMEDITEM	-
18	DBMS_XMLDOM.GETNEXTSIBLING	DBE_XMLDOM.GETNEXTSIBLING	-
19	DBMS_XMLDOM.GETNODENAME	DBE_XMLDOM.GETNODENAME	-
20	DBMS_XMLDOM.GETNODETYPE	DBE_XMLDOM.GETNODETYPE	-
21	DBMS_XMLDOM.GETTAGNAME	DBE_XMLDOM.GETTAGNAME	-
22	DBMS_XMLDOM.IMPORTNODE	DBE_XMLDOM.IMPORTNODE	-
23	DBMS_XMLDOM.ISNULL	DBE_XMLDOM.ISNULL	GaussDB: When the input parameter is of the DOMNODELIST type, an error is reported if the object does not exist in the hash table. Oracle Database: No error is reported.
24	DBMS_XMLDOM.ITEM	DBE_XMLDOM.ITEM	-
25	DBMS_XMLDOM.MAKENODE	DBE_XMLDOM.MAKENODE	GaussDB: This function cannot be directly returned as the function return value. Oracle Database: It is directly returned as the function return value.

No.	Oracle Database	GaussDB	Difference
26	DBMS_XMLDOM.NEWDOMDOCUMENT	DBE_XMLDOM.NEWDOMDOCUMENT	• The input parameter size in GaussDB must be less than 2 GB. Oracle Database: The size is the same as that of the CLOB type. • Currently, external DTD parsing is not supported in GaussDB. Oracle Database: External DTD can be parsed. • GaussDB: The default character set of doc created by newdomdocument is UTF-8. Oracle Database: It is generated based on the character set of the server. • GaussDB: Each doc parsed from the same xmltype instance is independent, and the modification of the doc does not affect the xmltype. Oracle Database: Each doc parsed from the same xmltype instance is not independent but associated. • GaussDB: The **version** column supports only **1.0**. If 1.0 to 1.9 are parsed, a warning is reported but the execution is normal. For versions later than 1.9, an error is reported. Oracle Database: No error is reported. • DTD verification difference between GaussDB and Oracle Database: **!ATTLIST to type (CHECK|check|Check) "Ch..."** reports an error because the default value **"Ch..."** is not an

No.	Oracle Database	GaussDB	Difference
			enumerated value in the brackets. However, Oracle Database does not report this error. **<!ENTITY baidu "www.baidu.com">..... &Baidu;&writer** reports an error. **Baidu** cannot correspond to **baidu** because the letters are case-sensitive. Oracle Database: No error is reported. • Namespace verification difference between GaussDB and Oracle Database: Undeclared namespace tags are parsed in GaussDB. Oracle Database: An error is reported.
27	DBMS_XMLDOM.SETATTRIBUTE	DBE_XMLDOM.SETATTRIBUTE	GaussDB: The attribute **key** cannot be null or an empty string. Oracle Database: The attribute **key** can be null or an empty string.
28	DBMS_XMLDOM.SETCHARSET	DBE_XMLDOM.SETCHARSET	Currently, the following character sets are supported in GaussDB: UTF-8, UTF-16, UCS-4, UCS-2, ISO-8859-1, ISO-8859-2, ISO-8859-3, ISO-8859-4, ISO-8859-5, ISO-8859-6, ISO-8859-7, ISO-8859-8, ISO-8859-9, ISO-2022-JP, Shift_JIS, EUC-JP, and ASCII. If you enter other character sets, an error is reported or garbled characters may be displayed.
29	DBMS_XMLDOM.SETDOCTYPE	DBE_XMLDOM.SETDOCTYPE	GaussDB: The total length of name, sysid, and pubid cannot exceed 32,500 bytes. Oracle Database: The maximum size is 32,767 bytes.

No.	Oracle Database	GaussDB	Difference
30	DBMS_XMLDOM.WRITETOBUFFER	DBE_XMLDOM.WRITETOBUFFER	• GaussDB: The writetobuffer output buffer is limited to less than 1 GB. Oracle Database: The maximum size is 32,767 bytes. • GaussDB: The output doc will contain the XML declaration version and encoding. Oracle Database: It is not contained unless being specified by users. • GaussDB: If the input parameter is of the domnode type and the node is converted from a doc, the output node contains the XML declaration version and encoding. Oracle Database: It is not contained unless being specified by users. • GaussDB: By default, XML files are output in the UTF-8 character set. Oracle Database: It is generated based on the database character set.

No.	Oracle Database	GaussDB	Difference
31	DBMS_XMLDOM.WRITETOCLOB	DBE_XMLDOM.WRITETOCLOB	• GaussDB: The writetoclob size cannot exceed 1 GB. Oracle Database: The supported size depends on the CLOB size. • GaussDB: The output doc will contain the XML declaration version and encoding. Oracle Database: It is not contained unless being specified by users. • GaussDB: If the input parameter is of the domnode type and the node is converted from a doc, the output node contains the XML declaration version and encoding. Oracle Database: It is not contained unless being specified by users. • GaussDB: By default, XML files are output in the UTF-8 character set. Oracle Database: It is generated based on the database character set.

No.	Oracle Database	GaussDB	Difference
32	DBMS_XMLDOM.WRITETOFILE	DBE_XMLDOM.WRITETOFILE	• GaussDB document input parameter. The length of **filename** cannot exceed 255 bytes. For details about **charset**, see the dbe_xmldom.setcharset API. Oracle Database: The length of **filename** is affected by the OS and is greater than 255 bytes. • GaussDB DOMNode input parameter. The length of **filename** cannot exceed 255 bytes. For details about **charset**, see the dbe_xmldom.setcharset API. Oracle Database: The length of **filename** is affected by the OS and is greater than 255 bytes. • GaussDB: This function adds content such as indentation to format the output. The output doc will contain the XML declaration version and encoding. If the input parameter is of the domnode type and the node is converted from a doc, the output node contains the XML declaration version and encoding. Oracle Database: It is not contained unless being specified by users. • GaussDB: If newdomdocument() is used to create a doc without parameters, no error is reported when **charset** is not specified. The UTF-8 character set is used by default. Oracle Database: An error is reported.

No.	Oracle Database	GaussDB	Difference
			• GaussDB: The filename must be in the path created in pg_directory. The backslash (\) in the filename will be converted to a slash (/). Only one slash (/) is allowed. The file name must be in the pg_directory_name/file_name format. Oracle Database: User input is not escaped.
33	DBMS_XMLDOM.GETNODEVALUE	DBE_XMLDOM.GETNODEVALUE	-
34	DBMS_XMLDOM.GETPARENTNODE	DBE_XMLDOM.GETPARENTNODE	-
35	DBMS_XMLDOM.HASCHILDNODES	DBE_XMLDOM.HASCHILDNODES	-
36	DBMS_XMLDOM.MAKEELEMENT	DBE_XMLDOM.MAKEELEMENT	-
37	DBMS_XMLDOM.SETNODEVALUE	DBE_XMLDOM.SETNODEVALUE	• GaussDB: Input parameter of **nodeValue**. You can enter an empty string or NULL, but the node value will not be changed. Oracle Database: If you enter an empty string or NULL, the node value is set to an empty string. • GaussDB: Input parameter of nodeValue. The escape character '&' is not supported. If the character string contains the escape character, the node value will be cleared. Oracle Database: Escape characters are supported.
38	DBMS_XMLDOM.GETELEMENTSBYTAGNAME	DBE_XMLDOM.GETELEMENTSBYTAGNAME	-

Table 13-15 DBMS_XMLPARSER compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_XMLPARSER.FREEPARSER	DBE_XMLPARSER.FREEPARSER	-
2	DBMS_XMLPARSER.GETDOCUMENT	DBE_XMLPARSER.GETDOCUMENT	-
3	DBMS_XMLPARSER.GETVALIDATIONMODE	DBE_XMLPARSER.GETVALIDATIONMODE	-
4	DBMS_XMLPARSER.NEWPARSER	Supported, with differences in DBE_XMLPARSER.NEWPARSER	The maximum number of parser objects in GaussDB is 16,777,215, and that in Oracle Database is about 100 million.

No.	Oracle Database	GaussDB	Difference
5	DBMS_XMLPARSER.PARSEBUFFER	Supported, with differences in DBE_XMLPARSER.PARSEBUFFER	1. Difference in parsing columns: Only UTF-8 is supported in terms of character encoding, and **version** can only be set to **1.0**. If versions 1.0 to 1.9 are parsed, a warning appears but the execution is normal. For versions later than 1.9, an error is reported. 2. Namespace validation difference: Undeclared namespace tags are parsed. However, Oracle Database reports an error. 3. Difference in parsing XML predefined entities: **'** and **"** are parsed and translated into **'** and **"**. However, predefined entities in Oracle Database are not translated into characters. 4. DTD validation differences: • **!ATTLIST to type (CHECK|check|Check) "Ch..."** reports an error because the default value **"Ch..."** is not an enumerated value in the brackets. However, Oracle Database does not report this error. • **<!ENTITY baidu "www.baidu.com">..... &Baidu;&writer** reports an error because the letters are case-sensitive. **Baidu** cannot correspond to **baidu**. However, Oracle Database does not report this error.

No.	Oracle Database	GaussDB	Difference
6	DBMS_XMLPARSER.PARSECLOB	Supported, with differences in DBE_XMLPARSER.PARSECLOB	1. PARSECLOB cannot parse CLOBs greater than or equal to 2 GB. 2. Difference in parsing columns: Only UTF-8 is supported in terms of character encoding, and **version** can only be set to **1.0**. If versions 1.0 to 1.9 are parsed, a warning appears but the execution is normal. For versions later than 1.9, an error is reported. 3. Namespace validation difference: Undeclared namespace tags are parsed. However, Oracle Database reports an error. 4. Difference in parsing XML predefined entities: **'** and **"** are parsed and translated into **'** and **"**. However, predefined entities in Oracle Database are not translated into characters. 5. DTD validation differences: • **!ATTLIST to type (CHECK|check|Check) "Ch..."** reports an error because the default value **"Ch..."** is not an enumerated value in the brackets. However, Oracle Database does not report this error. • **<!ENTITY baidu "www.baidu.com">..... &Baidu;&writer** reports an error because the letters are case-sensitive. **Baidu** cannot correspond to **baidu**. However, Oracle Database does not report this error.
7	DBMS_XMLPARSER.SETVALIDATIONMODE	DBE_XMLPARSER.SETVALIDATIONMODE	-

Table 13-16 DBMS_ILM compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_ILM.ADD_TO_ILM	Not supported.	-
2	DBMS_ILM.ARCHIVESTATENAME	Not supported.	-
3	DBMS_ILM.EXECUTE_ILM	Supported, with differences in DBE_ILM.EXECUTE_ILM	• The input parameter **schema** in GaussDB corresponds to **owner** in Oracle Database. • GaussDB does not support the operation of specifying **ilm_scope** (specifying multiple objects at a time).
4	DBMS_ILM.EXECUTE_ILM_TASK	Not supported.	-
5	DBMS_ILM.PREVIEW_ILM	Not supported.	-
6	DBMS_ILM.REMOVE_FROM_ILM	Not supported.	-
7	DBMS_ILM.STOP_ILM	DBE_ILM.STOP_ILM	-

Table 13-17 DBMS_ILM_ADMIN compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_ILM_ADMIN.CLEAR_HEAT_MAP_ALL	Not supported.	-
2	DBMS_ILM_ADMIN.CLEAR_HEAT_MAP_TABLE	Not supported.	-

No.	Oracle Database	GaussDB	Difference
3	DBMS_ILM_ADMIN.CUSTOMIZE_ILM	Supported, with differences in DBE_ILM_ADMIN.CUSTOMIZE_ILM	The feature parameters corresponding to the values of input parameters are different. The value of **param** in GaussDB can be **1**, **2**, **7**, **11**, **12**, **13**, **14**, or **15**. When the value of **param** in GaussDB is **14**, the corresponding feature parameter is **WIND_DURATION** which is used to control the execution window duration in automatic scheduling. When the value of **param** in Oracle Database is **14**, the corresponding feature parameter is **AUTO_OPTIMIZE_INACTIVITY_THRESHOLD** which indicates the inactive duration of the ADO.
4	DBMS_ILM_ADMIN.DISABLE_ILM	DBE_ILM_ADMIN.DISABLE_ILM	-
5	DBMS_ILM_ADMIN.ENABLE_AUTO_OPTIMIZE	Not supported.	-
6	DBMS_ILM_ADMIN.ENABLE_ILM	DBE_ILM_ADMIN.ENABLE_ILM	-
7	DBMS_ILM_ADMIN. IGNORE_AUTO_OPTIMIZE_ CRITERIA	Not supported.	-
8	DBMS_ILM_ADMIN.SET_HEAT_MAP_ALL	Not supported.	-
9	DBMS_ILM_ADMIN.SET_HEAT_MAP_START	Not supported.	-
10	DBMS_ILM_ADMIN.SET_HEAT_MAP_TABLE	Not supported.	-

Table 13-18 DBMS_COMPRESSION compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_COMPRESSION.GET_COMPRESSION_RATIO	Supported, with differences in DBE_COMPRESSION.GET_COMPRESSION_RATIO	• GaussDB cannot obtain compression rates of LOBs. • For obtaining a compression rate of a single object: – The value of the input parameter **comptype** in GaussDB can only be **1** (uncompressed) or **2** (advanced compression) while Oracle Database also supports values such as **1024** and **2048**. – The value of the input parameter **objtype** in GaussDB can only be **1** (table object) while Oracle Database also supports the value **2** (index object). – Oracle Database uses the **subset_numrows** parameter to directly determine the number of rows to be sampled (that is, the value of the parameter). GaussDB uses **sample_ratio** (sampling rate) to indirectly determine the number of rows to be sampled.
2	DBMS_COMPRESSION.GET_COMPRESSION_TYPE	Supported, with differences in DBE_COMPRESSION.GET_COMPRESSION_TYPE	• Oracle Database uses a row ID to specify the row whose compression type is to be obtained, while GaussDB uses a CTID to specify the row. • The value of **comptype** is returned. The value difference is the same as that of GET_COMPRESSION_RATIO.

Table 13-19 DBMS_HEAT_MAP compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_HEAT_MAP.BLOCK_HEAT_MAP	Not supported.	-
2	DBMS_HEAT_MAP.EXTENT_HEAT_MAP	Not supported.	-
3	DBMS_HEAT_MAP.OBJECT_HEAT_MAP	Not supported.	-
4	DBMS_HEAT_MAP.SEGMENT_HEAT_MAP	Not supported.	-
5	DBMS_HEAT_MAP.TABLESPACE_HEAT_MAP	Not supported.	-
6	Not supported.	DBE_HEAT_MAP.ROW_HEAT_MAP	For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_HEAT_MAP" in *Developer Guide*.

Table 13-20 DBMS_DESCRIBE compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_DESCRIBE.DESCRIBE_PROCEDURE	Supported, with differences in DBE_DESCRIBE.DESCRIBE_PROCEDURE	• The **datatype** parameter is different from the O database. GaussDB returns the OID of the data type, and the O database returns the ID of the data type in the O database. • The **datalength**, **dataprecision**, and **scale** parameters are set to **0** because type constraints (such as number (7,2) and varchar2(20)) cannot be retained when GaussDB creates stored procedures or functions. Oracle Database can use the %type method to obtain constrained data types. • For details about the behavior differences, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_DESCRIBE" in *Developer Guide*.

Table 13-21 DBMS_STATS compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_STATS.ALTER_STATS_HISTORY_RETENTION	Not supported.	-
2	DBMS_STATS.CANCEL_ADVISOR_TASK	Not supported.	-
3	DBMS_STATS.CONFIGURE_ADVISOR_FILTER	Not supported.	-
4	DBMS_STATS.CONFIGURE_ADVISOR_OBJ_FILTER	Not supported.	-

No.	Oracle Database	GaussDB	Difference
5	DBMS_STATS.CONFIGURE_ADVISOR_OPR_FILTER	Not supported.	-
6	DBMS_STATS.CONFIGURE_ADVISOR_RULE_FILTER	Not supported.	-
7	DBMS_STATS.CREATE_ADVISOR_TASK	Not supported.	-
8	DBMS_STATS.CONVERT_RAW_VALUE	Not supported.	-
9	DBMS_STATS.CONVERT_RAW_VALUE_NVARCHAR	Not supported.	-
10	DBMS_STATS.CONVERT_RAW_VALUE_ROWID	Not supported.	-
11	DBMS_STATS.COPY_TABLE_STATS	Not supported.	-
12	DBMS_STATS.CREATE_EXTENDED_STATS	Not supported.	-
13	DBMS_STATS.CREATE_STAT_TABLE	DBE_STATS.CREATE_STAT_TABLE	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*.

No.	Oracle Database	GaussDB	Difference
14	DBMS_STATS.DELETE_COLUMN_STATS	DBE_STATS.DELETE_COLUMN_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • In GaussDB, this API can be used to set expression statistics, but **tabname** must be set to an index name corresponding to the expression.
15	DBMS_STATS.DELETE_DATABASE_PREFS	Not supported.	-
16	DBMS_STATS.DELETE_DATABASE_STATS	Not supported.	-
17	DEDBMS_STATS.DELETE_DICTIONARY_STATS	Not supported.	-
18	DBMS_STATS.DELETE_FIXED_OBJECTS_STATS	Not supported.	-
19	DBMS_STATS.DELETE_INDEX_STATS	DBE_STATS.DELETE_INDEX_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*.
20	DBMS_STATS.DELETE_PENDING_STATS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
21	DBMS_STATS.DELETE_PROCESSING_RATE	Not supported.	-
22	DBMS_STATS.DELETE_SCHEMA_PREFS	Not supported.	-
23	DBMS_STATS.DELETE_SCHEMA_STATS	DBE_STATS.DELETE_SCHEMA_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*.
24	DBMS_STATS.DELETE_SYSTEM_STATS	Not supported.	-
25	DBMS_STATS.DELETE_TABLE_PREFS	Not supported.	-
26	DBMS_STATS.DELETE_TABLE_STATS	DBE_STATS.DELETE_TABLE_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*.
27	DBMS_STATS.DIFF_TABLE_STATS_IN_HISTORY	Not supported.	-
28	DBMS_STATS.DIFF_TABLE_STATS_IN_PENDING	Not supported.	-
29	DBMS_STATS.DIFF_TABLE_STATS_IN_STATTAB	Not supported.	-

No.	Oracle Database	GaussDB	Difference
30	DBMS_STATS.DROP_ADVISOR_TASK	Not supported.	-
31	DBMS_STATS.DROP_EXTENDED_STATS	Not supported.	-
32	DBMS_STATS.DROP_STAT_TABLE	DBE_STATS.DROP_STAT_TABLE	-
33	DBMS_STATS.EXECUTE_ADVISOR_TASK	Not supported.	-
34	DBMS_STATS.EXPORT_COLUMN_STATS	DBE_STATS.EXPORT_COLUMN_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • The exported column-level statistics are consistent with those in the pg_statistic catalog. Multiple columns are consistent with those in the pg_statistic_ext catalog. • Index expression statistics can be exported. **tabname** must be set to an index name, and **colname** must be set to an index expression name. • Permission: You must have the analyze permission to query tables and the siud permission on the **stattab** table.
35	DBMS_STATS.EXPORT_DATABASE_PREFS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
36	DBMS_STATS.EXPORT_DATABASE_STATS	Not supported.	-
37	DBMS_STATS.EXPORT_DICTIONARY_STATS	Not supported.	-
38	DBMS_STATS.EXPORT_FIXED_OBJECTS_STATS	Not supported.	-
39	DBMS_STATS.EXPORT_INDEX_STATS	DBE_STATS.EXPORT_INDEX_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • In the **stattab** table, the exported table-level and partition-level statistics are **numrows**, **numblocks**, and **relallvisible**, which correspond to **reltuples**, **relpages**, and **relallvisible** in the pg_class and pg_partition system catalogs, respectively. • Permission: You must have the analyze permission to query tables and the siud permission on the **stattab** table.
40	DBMS_STATS.EXPORT_PENDING_STATS	Not supported.	-
41	DBMS_STATS.EXPORT_SCHEMA_PREFS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
42	DBMS_STATS.EXPORT_SCHEMA_STATS	DBE_STATS.EXPORT_SCHEMA_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • In the **stattab** table, the exported table-level and partition-level statistics are **numrows**, **numblocks**, and **relallvisible**, which correspond to **reltuples**, **relpages**, and **relallvisible** in the pg_class and pg_partition system catalogs, respectively. The column-level statistics of the exported table are consistent with those of the pg_statistic and pg_statistic_ext catalogs. • Permission: You must have the siud permission on the **stattab** table.
43	DBMS_STATS.EXPORT_SYSTEM_STATS	Not supported.	-
44	DBMS_STATS.EXPORT_TABLE_PREFS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
45	DBMS_STATS.EXPORT_TABLE_STATS	DBE_STATS.EXPORT_TABLE_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • In the **stattab** table, the exported table-level and partition-level statistics are **numrows**, **numblocks**, and **relallvisible**, which correspond to **reltuples**, **relpages**, and **relallvisible** in the pg_class and pg_partition system catalogs, respectively. The column-level statistics exported in cascading mode are consistent with those in the pg_statistic and pg_statistic_ext catalogs. • Permission: You must have the analyze permission to query tables and the siud permission on the **stattab** table.
46	DBMS_STATS.FLUSH_DATABASE_MONITORING_INFO	Not supported.	-
47	DBMS_STATS.GATHER_DATABASE_STATS	Not supported.	-
48	DBMS_STATS.GATHER_DICTIONARY_STATS	Not supported.	-
49	DBMS_STATS.GATHER_FIXED_OBJECTS_STATS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
50	DBMS_STATS.GATHER_INDEX_STATS	Not supported.	-
51	DBMS_STATS.GATHER_PROCESSING_RATE	Not supported.	-
52	DBMS_STATS.GATHER_SCHEMA_STATS	Not supported.	-
53	DBMS_STATS.GATHER_SYSTEM_STATS	Not supported.	-
54	DBMS_STATS.GATHER_TABLE_STATS	Not supported.	-
55	DBMS_STATS.GENERATE_STATS	Not supported.	-
56	DBMS_STATS.GET_ADVISOR_OPR_FILTER	Not supported.	-
57	DBMS_STATS.GET_ADVISOR_RECS	Not supported.	-
58	DBMS_STATS.GET_COLUMN_STATS	Not supported.	-
59	DBMS_STATS.GET_INDEX_STATS	Not supported.	-
60	DBMS_STATS.GET_PARAM	Not supported.	-
61	DBMS_STATS.GET_PREFS	Not supported.	-
62	DBMS_STATS.GET_STATS_HISTORY_AVAILABILITY	DBE_STATS.GET_STATS_HISTORY_AVAILABILITY	GaussDB queries the collection time of the earliest historical statistics in the entire database.
63	DBMS_STATS.GET_STATS_HISTORY_RETENTION	DBE_STATS.GET_STATS_HISTORY_RETENTION	-
64	DBMS_STATS.GET_SYSTEM_STATS	Not supported.	-
65	DBMS_STATS.GET_TABLE_STATS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
66	DBMS_STATS.IMPLEMENT_ADVISOR_TASK	Not supported.	-
67	DBMS_STATS.IMPORT_COLUMN_STATS	DBE_STATS.IMPORT_COLUMN_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • The exported single-column **col** statistics are the same as those in the pg_statistic catalog. The exported multi-column **ext-col** statistics are consistent with those in the pg_statistic_ext catalog. • Index expression statistics can be imported. **tabname** must be set to an index name, and **colname** must be set to an index expression name. • Permission: You must have the analyze permission to query tables and the siud permission on the **stattab** table.
68	DBMS_STATS.IMPORT_DATABASE_PREFS	Not supported.	-
69	DBMS_STATS.IMPORT_DATABASE_STATS	Not supported.	-
70	DBMS_STATS.IMPORT_DICTIONARY_STATS	Not supported.	-
71	DBMS_STATS.IMPORT_FIXED_OBJECTS_STATS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
72	DBMS_STATS.IMPORT_INDEX_STATS	DBE_STATS.IMPORT_INDEX_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • In the **stattab** table, the imported table-level and partition-level statistics are **numrows**, **numblocks**, and **relallvisible**, which correspond to **reltuples**, **relpages**, and **relallvisible** in the pg_class and pg_partition system catalogs, respectively. • Permission: You must have the analyze permission to query tables and the siud permission on the **stattab** table.
73	DBMS_STATS.IMPORT_SCHEMA_PREFS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
74	DBMS_STATS.IMPORT_SCHEMA_STATS	DBE_STATS.IMPORT_SCHEMA_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • In the **stattab** table, the imported table-level and partition-level statistics are **numrows**, **numblocks**, and **relallvisible**, which correspond to **reltuples**, **relpages**, and **relallvisible** in the pg_class and pg_partition system catalogs, respectively. The column-level statistics of the imported table are consistent with those of the pg_statistic and pg_statistic_ext catalogs. • Permission: You must have the siud permission on the **stattab** table.
75	DBMS_STATS.IMPORT_SYSTEM_STATS	Not supported.	-
76	DBMS_STATS.IMPORT_TABLE_PREFS	Not supported.	-

No.	Oracle Database	GaussDB	Difference
77	DBMS_STATS.IMPORT_TABLE_STATS	DBE_STATS.IMPORT_TABLE_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • In the **stattab** table, the imported table-level and partition-level statistics are **numrows**, **numblocks**, and **relallvisible**, which correspond to **reltuples**, **relpages**, and **relallvisible** in the pg_class and pg_partition system catalogs, respectively. The column-level statistics imported in cascading mode are consistent with those in the pg_statistic and pg_statistic_ext catalogs. • Permission: You must have the analyze permission to query tables and the siud permission on the **stattab** table.
78	DBMS_STATS.INTERRUPT_ADVISOR_TASK	Not supported.	-
79	DBMS_STATS.LOCK_PARTITION_STATS	DBE_STATS.LOCK_PARTITION_STATS	In GaussDB, **ownname** must be set to a schema name.
80	DBMS_STATS.LOCK_SCHEMA_STATS	DBE_STATS.LOCK_SCHEMA_STATS	In GaussDB, **ownname** must be set to a schema name.
81	DBMS_STATS.LOCK_TABLE_STATS	DBE_STATS.LOCK_TABLE_STATS	In GaussDB, **ownname** must be set to a schema name.
82	DBMS_STATS.MERGE_COL_USAGE	Not supported.	-

No.	Oracle Database	GaussDB	Difference
83	DBMS_STATS.PREPARE_COLUMN_VALUES	Not supported.	-
84	DBMS_STATS.PREPARE_COLUMN_VALUES_ROWID	Not supported.	-
85	DBMS_STATS.PUBLISH_PENDING_STATS	Not supported.	-
86	DBMS_STATS.PURGE_STATS	DBE_STATS.PURGE_STATS	-
87	DBMS_STATS.REMAP_STAT_TABLE	Not supported.	-
88	DBMS_STATS.REPORT_ADVISOR_TASK	Not supported.	-
89	DBMS_STATS.REPORT_COL_USAGE	Not supported.	-
90	DBMS_STATS.REPORT_GATHER_AUTO_STATS	Not supported.	-
91	DBMS_STATS.REPORT_GATHER_DATABASE_STATS	Not supported.	-
92	DBMS_STATS.REPORT_GATHER_DICTIONARY_STATS	Not supported.	-
93	DBMS_STATS.REPORT_GATHER_FIXED_OBJ_STATS	Not supported.	-
94	DBMS_STATS.REPORT_GATHER_SCHEMA_STATS	Not supported.	-
95	DBMS_STATS.REPORT_STATS_OPERATIONS	Not supported.	-
96	DBMS_STATS.RESET_ADVISOR_TASK	Not supported.	-
97	DBMS_STATS.RESET_COL_USAGE	Not supported.	-

No.	Oracle Database	GaussDB	Difference
98	DBMS_STATS.RESET_GLOBAL_PREF_DEFAULTS	Not supported.	-
99	DBMS_STATS.RESET_PARAM_DEFAULTS	Not supported.	-
100	DBMS_STATS.RESTORE_DICTIONARY_STATS	Not supported.	-
101	DBMS_STATS.RESTORE_FIXED_OBJECTS_STATS	Not supported.	-
102	DBMS_STATS.RESTORE_SCHEMA_STATS	DBE_STATS.RESTORE_SCHEMA_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*.
103	DBMS_STATS.RESTORE_SYSTEM_STATS	Not supported.	-
104	DBMS_STATS.RESTORE_TABLE_STATS	DBE_STATS.RESTORE_TABLE_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*.
105	DBMS_STATS.RESUME_ADVISOR_TASK	Not supported.	-

No.	Oracle Database	GaussDB	Difference
106	DBMS_STATS.SCRIPT_ADVISOR_TASK	Not supported.	-
107	DBMS_STATS.SEED_COL_USAGE	Not supported.	-
108	DBMS_STATS.SET_ADVISOR_TASK_PARAMETER	Not supported.	-
109	DBMS_STATS.SET_COLUMN_STATS	DBE_STATS.SET_COLUMN_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*.
110	DBMS_STATS.SET_DATABASE_PREFS	Not supported.	-
111	DBMS_STATS.SET_GLOBAL_PREFS	Not supported.	-
112	DBMS_STATS.SET_INDEX_STATS	DBE_STATS.SET_INDEX_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • The **relallvisible** input parameter is added to GaussDB.
113	DBMS_STATS.SET_PARAM	Not supported.	-
114	DBMS_STATS.SET_PROCESSING_RATE	Not supported.	-

No.	Oracle Database	GaussDB	Difference
115	DBMS_STATS.SET_SCHEMA_PREFS	Not supported.	-
116	DBMS_STATS.SET_SYSTEM_STATS	Not supported.	-
117	DBMS_STATS.SET_TABLE_PREFS	Not supported.	-
118	DBMS_STATS.SET_TABLE_STATS	DBE_STATS.SET_TABLE_STATS	• In GaussDB, **ownname** must be set to a schema name. • GaussDB supports only some input parameter functions. For details, see "Stored Procedure > Advanced Package > Secondary Encapsulation Interfaces (Recommended) > DBE_STATS" in *Developer Guide*. • The **relallvisible** input parameter is added to GaussDB.
119	DBMS_STATS.SHOW_EXTENDED_STATS_NAME	Not supported.	-
120	DBMS_STATS.TRANSFER_STATS	Not supported.	-
121	DBMS_STATS.UNLOCK_PARTITION_STATS	DBE_STATS.UNLOCK_PARTITION_STATS	In GaussDB, **ownname** must be set to a schema name.
122	DBMS_STATS.UNLOCK_SCHEMA_STATS	DBE_STATS.UNLOCK_SCHEMA_STATS	In GaussDB, **ownname** must be set to a schema name.
123	DBMS_STATS.UNLOCK_TABLE_STATS	DBE_STATS.UNLOCK_TABLE_STATS	In GaussDB, **ownname** must be set to a schema name.
124	DBMS_STATS.UPGRADE_STAT_TABLE	Not supported.	-

Table 13-22 DBMS_XMLGEN compatibility

No.	Oracle Database	GaussDB	Difference
1	DBMS_XMLGEN.CONVERT	DBE_XMLGEN.CONVERT	-
2	DBMS_XMLGEN.NEWCONTEXT	DBE_XMLGEN.NEWCONTEXT	-
3	DBMS_XMLGEN.NEWCONTEXTFROMHIERARCHY	DBE_XMLGEN.NEWCONTEXTFROMHIERARCHY	• The maximum depth of recursive XML files generated by GaussDB cannot exceed 50 million layers. • XML files generated by the CONNECT BY statement in Oracle Database's newcontextfromhierarchy method contain XML headers. However, the directly constructed data does not contain the XML header. In GaussDB, the files contain XML headers.
4	DBMS_XMLGEN.SETCONVERTSPECIALCHARS	DBE_XMLGEN.SETCONVERTSPECIALCHARS	-
5	DBMS_XMLGEN.SETNULLHANDLING	DBE_XMLGEN.SETNULLHANDLING	-
6	DBMS_XMLGEN.SETROWSETTAG	DBE_XMLGEN.SETROWSETTAG	-
7	DBMS_XMLGEN.SETROWTAG	DBE_XMLGEN.SETROWTAG	-
8	DBMS_XMLGEN.USENULLATTRIBUTEINDICATOR	DBE_XMLGEN.USENULLATTRIBUTEINDICATOR	-
9	DBMS_XMLGEN.USEITEMTAGSFORCOLL	DBE_XMLGEN.USEITEMTAGSFORCOLL	-
10	DBMS_XMLGEN.GETNUMROWSPROCESSED	DBE_XMLGEN.GETNUMROWSPROCESSED	-
11	DBMS_XMLGEN.SETMAXROWS	DBE_XMLGEN.SETMAXROWS	-

No.	Oracle Database	GaussDB	Difference
12	DBMS_XMLGEN.SETSKIPROWS	DBE_XMLGEN.SETSKIPROWS	-
13	DBMS_XMLGEN.RESTARTQUERY	DBE_XMLGEN.RESTARTQUERY	• GaussDB: The updated data is invisible after the RESTARTQUERY method is called. • Oracle Database: The updated data is visible after the RESTARTQUERY method is called.
14	DBMS_XMLGEN.GETXMLTYPE	DBE_XMLGEN.GETXMLTYPE	-
15	DBMS_XMLGEN.GETXML	DBE_XMLGEN.GETXML	-
16	DBMS_XMLGEN.CLOSECONTEXT	DBE_XMLGEN.CLOSECONTEXT	-

