



中华人民共和国国家标准

GB/T 12991.1—2008/ISO/IEC 9075-1:2003
代替 GB/T 12991—1991

信息技术 数据库语言 SQL 第 1 部分：框架

Information technology—Database languages—SQL—
Part 1: framework

(ISO/IEC 9075-1:2003, Information technology—
Database languages—SQL—Part 1: Framework(SQL/Framework), IDT)

2008-07-16 发布

2008-12-01 实施

中华人民共和国国家质量监督检验检疫总局 发布
中国国家标准化管理委员会

目 次

前言	V
引言	VI
1 范围	1
2 规范性引用文件	1
3 术语的定义和使用	1
3.1 定义	1
3.1.1 本标准中提供的定义	1
3.2 术语的使用	3
3.3 提示性内容	3
4 概念	3
4.1 防止误解的说明	3
4.2 SQL 环境及其组成	4
4.2.1 SQL 环境	4
4.2.2 SQL 代理	4
4.2.3 SQL 实现	4
4.2.4 SQL 客户模块	4
4.2.5 用户标识符	5
4.2.5a 角色	5
4.2.6 用户映射概念	5
4.2.7 例程映射概念	5
4.2.8 目录和模式	5
4.2.9 外部服务器和描述符	6
4.2.10 外部数据封装器和描述符	6
4.2.11 SQL 数据	6
4.3 表	6
4.4 SQL 数据类型	6
4.4.1 一般数据类型信息	6
4.4.2 空值	7
4.4.3 预定义类型	7
4.4.4 构造原子类型	8
4.4.5 构造复合类型	8
4.5 场所及场所上的操作	8
4.5.1 场所	8
4.5.2 赋值	8
4.5.3 可空性	8
4.6 SQL 模式对象	8
4.6.1 一般 SQL 模式对象信息	8
4.6.2 与字符集相关的描述符	9

4.6.3	域及其组成	9
4.6.4	用户定义类型	10
4.6.5	相异类型	10
4.6.6	基表及组成	10
4.6.7	视图定义	11
4.6.8	断言	11
4.6.9	SQL 服务器模块	11
4.6.10	模式例程	11
4.6.11	序列生成器	11
4.6.12	特权	11
4.7	完整性约束和约束检查	11
4.7.1	约束检查	11
4.7.2	确定性和约束	12
4.8	SQL 代理和 SQL 服务器之间的通信	12
4.8.1	宿主语言	12
4.8.2	参数传递和数据类型对应性	12
4.8.3	描述符区	13
4.8.4	诊断信息	13
4.8.5	SQL 事务	13
4.9	模块	14
4.10	例程	14
4.10.1	一般例程信息	14
4.10.2	类型保留函数	14
4.11	SQL 语句	14
4.11.1	SQL 语句分类	14
4.11.2	按功能分类的 SQL 语句	15
5	GB/T 12991 中的各部分说明	15
5.1	概述	15
5.2	GB/T 12991. 1: 框架	15
5.3	GB/T 12991. 2: 基础	16
5.3.1	GB/T 12991. 2 规定的数据类型	16
5.3.2	表	16
5.3.3	绑定方法	16
5.3.4	GB/T 12991. 2 中规定的 SQL 语句	16
5.4	GB/T 12991. 3: 调用层接口	17
5.5	GB/T 12991. 4: 持久存储模块	17
5.5.1	GB/T 12991. 4 中规定的 SQL 语句	17
5.6	GB/T 12991. 9: 外部数据管理	17
5.7	GB/T 12991. 10: 对象语言绑定	18
5.8	GB/T 12991. 11: 信息和定义模式	18
5.9	GB/T 12991. 13: 使用 Java 程序设计语言的 SQL 例程和类型	18
5.10	GB/T 12991. 14: 与 XML 相关的规范	18
6	GB/T 12991 中其他各部分使用的记法和约定	18

6.1	取自 ISO/IEC 10646 中的记法	18
6.2	本标准中使用的记法	18
6.3	约定	19
6.3.1	句法元素说明	19
6.3.2	信息模式和定义模式的说明	20
6.3.3	术语的使用	20
6.3.4	描述符	23
6.3.5	GB/T 12991 中各部分的关系	23
6.3.5a	作为子例程使用的条款	26
6.3.6	索引排版格式	26
6.3.7	特征 ID 和特征名	26
6.4	数据库语言 SQL 的对象标识符	27
7	GB/T 12991 中各部分的附录	28
7.1	实现定义的元素	28
7.2	实现相关的元素	28
7.3	不推荐的特征	28
7.4	与以前版本的不兼容性	28
8	符合性	28
8.1	最小符合性	28
8.2	部分的符合性	28
8.3	特征的符合性	28
8.4	SQL 包的符合性	29
8.4.1	增强日期时间设施	29
8.4.2	增强完整性管理	29
8.4.3	PSM	30
8.4.4	基本对象支持	30
8.4.5	增强对象支持	30
8.4.6	主动数据库	30
8.4.7	OLAP	30
8.4.8	扩展和选项	30
8.5	SQL 标记符	31
8.6	符合性声明	32
8.6.1	对 SQL 应用的要求	32
8.6.2	对 SQL 实现的要求	32
附录 A (资料性附录)	SQL 国际标准的维护和解释	33
附录 B (资料性附录)	实现定义的元素	34
附录 C (资料性附录)	实现相关的元素	35
附录 NA (资料性附录)	GB/T 12991 标准发展历程	36
参考文献		37

前 言

GB/T 12991 在《信息技术 数据库语言 SQL》总标题下分为 9 个部分：

- 第 1 部分：框架；
- 第 2 部分：基础；
- 第 3 部分：调用层接口；
- 第 4 部分：持久存储模块；
- 第 9 部分：外部数据管理；
- 第 10 部分：对象语言绑定；
- 第 11 部分：信息和定义模式；
- 第 13 部分：使用 Java 程序设计语言的 SQL 例程和类型；
- 第 14 部分：与 XML 相关的规范。

本部分为 GB/T 12991 的第 1 部分。本部分等同采用 ISO/IEC 9075-1:2003《信息技术 数据库语言 SQL 第 1 部分：框架(SQL/框架)》、ISO/IEC 9075-1:2003/cor. 1:2005 技术勘误表 1、ISO/IEC 9075-1:2003/cor. 2:2007 技术勘误表 2，并做了如下编辑性修改：

- 改正 ISO/IEC 9075-1:2003 标准中出现的印刷错误，将原英文版 4.6.2.2“理序”一节第三行中出现的 14561 更改为 14651，此改动在正文中的页边空白处用垂直线(|)标识。
- 将 ISO/IEC 9075-1:2003 对应的两个技术勘误 ISO/IEC 9075-1:2003/Cor. 1:2005 和 ISO/IEC 9075-1:2003/Cor. 2:2007 并入到 GB/T 12991.1:2003 的文本中，并在正文中的页边空白处用垂直双线(=)标识。
- 用“本标准”代替了“本国际标准”。
- 在第 1 部分“框架”后增加资料性附录 NA，说明 GB/T 12991 标准的发展历程。
- 删除 ISO/IEC 9075-1:2003 国际标准的前言，增加国家标准对应的前言。
- 第 2 章“规范性引用文件”中对 ISO/IEC 10646 标准的引用，由原来的引用 ISO/IEC 10646-1:2000 和 ISO/IEC 10646-2:2001 两个标准改为引用 ISO/IEC 10646:2003 一个标准。

本部分与 GB/T 12991 的其他部分共同代替标准 GB/T 12991—1991。本部分与前一版本的主要技术变化在附录 NA 中有详细介绍。

本部分的附录 A、附录 B、附录 C、附录 NA 为资料性附录。

本部分由中华人民共和国信息产业部提出。

本部分由全国信息技术标准化技术委员会归口。

本部分起草单位：中国电子技术标准化研究所、武汉达梦数据库有限公司。

本部分主要起草人：冯玉才、吴恒山、谢美意、班鹏新、李晨阳、李海波、赵菁华、卫凤林、余云涛。

本部分所代替标准的历次版本发布情况为：

- GB/T 12991—1991。

引 言

GB/T 12991 本部分的内容组织如下：

- a) 第1章“范围”，说明 GB/T 12991 本部分所涉及的范围。
- b) 第2章“规范性引用文件”，标识 GB/T 12991 本部分引用的附加标准，通过对附加标准的引用构成 GB/T 12991 本部分的补充条款。
- c) 第3章“术语的定义和使用”，定义 GB/T 12991 中所使用的术语。
- d) 第4章“概念”，描述 GB/T 12991 中使用的概念。
- e) 第5章“GB/T 12991 中的各部分说明”，依据第4章“概念”中所描述的概念，概括 GB/T 12991 各部分的基本内容。
- f) 第6章“GB/T 12991 中其他各部分使用的记法和约定”，定义 GB/T 12991 中其他部分使用的记法和约定。
- g) 第7章“GB/T 12991 中各部分的附录”，描述 GB/T 12991 中其他部分相关的附录内容。
- h) 第8章“符合性”，说明对 GB/T 12991 中全部或部分内容满足符合性所需达到的要求。
- i) 附录 A“SQL 国际标准的维护和解释”，属于资料性附录。该附录描述维护和解释 ISO/IEC 9075 标准的正式程序。
- j) 附录 B“实现定义的元素”，属于资料性附录。该附录列出了 GB/T 12991 本部分内容的语法、含义、返回结果、对 SQL 数据和/或模式的影响或任何其他行为是部分还是全部是实现定义的那些特征。
- k) 附录 C“实现相关的元素”，属于资料性附录。该附录列出了 GB/T 12991 本部分内容的语法、含义、返回结果、对 SQL 数据和/或模式的影响或任何其他行为是部分还是全部是实现相关的那些特征。
- l) 附录 NA“GB/T 12991 标准发展历程”属于资料性附录。该附录列出了 GB/T 12991 的发展历程及各版本之间的差异。
- m) 参考文献介绍了一些 Unicode 标准及相关链接。

信息技术 数据库语言 SQL

第 1 部分:框架

1 范围

GB/T 12991 的本部分描述 GB/T 12991 其他部分中所使用的概念性框架,规定了 SQL 语法和 SQL 实现的处理语句的结果。

本部分还定义了 GB/T 12991 其他部分使用的术语和记法。

2 规范性引用文件

下列文件中的条款通过 GB/T 12991 的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 12991 的所有部分(ISO/IEC 9075:2003, IDT)

GB/T 16262.1—2006 信息技术 抽象语法记法一(ASN.1) 第 1 部分:基本记法规范(ISO/IEC 8824-1:2002, IDT)

ISO/IEC 10646:2003 信息技术 通用多八位编码字符集(UCS)

ISO/IEC 14651:2001 信息技术 国际字符串排序和比较 比较字符串和描述公用模板可裁剪的排序方法

3 术语的定义和使用

3.1 定义

3.1.1 本标准中提供的定义

下列定义适用于 GB/T 12991 的本部分及其他部分。

3.1.1.1

原子的 atomic

不能被进一步分割的。

3.1.1.2

编译单元 compilation unit

可能由一个或多个子程序组成的一段可执行代码。

3.1.1.3

数据类型 data type

可表示的值的集合。

3.1.1.4

描述符 descriptor

SQL 对象的编码描述。它包括一个符合 SQL 实现所要求的对象的全部信息。

3.1.1.5

某些 SQL 对象名的完全限定 fully qualified of a name of some SQL object

带有显式说明的所有可选部分。

注 1: 完全限定名对于唯一标识一个对象并不一定是必需的。例如,虽然由目录名、模式名和专用名组成的完全限定专用名能唯一标识一个例程,但并非必须要完全限定例程名。

3.1.1.6

标识符 identifier

标识事物的方式。

3.1.1.7

识别 identify

无歧义性地指认事物。

3.1.1.8

实现定义的 implementation-defined

SQL 实现之间可能不同,而由每个特定 SQL 实现的实现者进行规定。

3.1.1.9

实现相关的 implementation-dependent

SQL 实现之间可能不同,但 GB/T 12991 中没有规定的、且不要求任何特定 SQL 实现的实现者进行规定的内容。

3.1.1.10

(一个值的)实例 instance (of a value)

一个值的物理表示。每个实例都确切地存在于一个场所。每个实例都有其数据类型,该数据类型就是其值的数据类型。

3.1.1.11

空值 null value

用来表示任何数据值暂缺的特殊值。

3.1.1.12

对象(如“X 对象”) object (as in “X object”)

任何事物。一个 X 对象是某个 X 的组成部分或与其相关,不能脱离 X 独立存在的事物。例如,一个 SQL 对象必须存在于 SQL 上下文中;一个 SQL 模式对象必须存在于某个 SQL 模式中。

3.1.1.13

持久的 persistent

除非有意撤销,始终持续存在。引用和级联动作视为有意的动作。SQL 事务或 SQL 会话偶然发生的终止动作不视为有意的动作。

3.1.1.14

(对象的)性质 property (of an object)

有关某个对象的属性、质量和特征。

3.1.1.15

行 row

(字段名,值)对的序列,每个值的数据类型由行类型来指定。

3.1.1.16

(标准的)范围 scope (of a standard)

本标准中的条款。该条款定义本标准的主题和所涉及的内容,从而指出本标准或其中某个部分的适用限制。

3.1.1.17

(名字或声明的)作用域 scope (of a name or declaration)

使得名字或声明有效的一个或多个 BNF 非终结符。

3.1.1.18

(引用类型的)作用域 scope (of a reference type)

引用类型的值所引用的表。

3.1.1.19

序列 sequence

未必相异的对象的有序集合。

3.1.1.20

场所 site

某个指定数据类型(或它的子类型)的值的实例占用的地方。

3.1.1.21

规模化项 sizing item

SQL 实现或配置文件中实现定义项的值。

3.1.1.22

SQL 连接 SQL-connection

SQL 客户与 SQL 服务器之间的联系。

3.1.1.23

SQL 环境 SQL-environment

SQL 数据存在和 SQL 语句执行的上下文环境。

3.1.1.24

SQL 实现 SQL-implementation

处理 SQL 语句的处理器。符合 SQL 实现是指满足第 8 章“符合性”中定义的要求的 SQL 实现。

3.1.1.25

SQL 会话 SQL-session

在一个 SQL 连接上,单个 SQL 代理的单个用户执行一个连续 SQL 语句序列的上下文环境。

3.1.1.26

SQL 语句 SQL-statement

符合 GB/T 12991 中规定的格式和语法规则的字符串。

3.1.1.27

表 table

具有一个或多个列的有序汇集的行的无序汇集。每一列有一个名称和一个数据类型。对于每一列,每一行只有一个属于该列的数据类型的值。

3.2 术语的使用

GB/T 12991 中的概念是依据对象并按照通常的意义来描述的。

一个对象可以看作是它所依赖的那个对象的组成部分。如果一个对象不复存在,则依赖于它的每个对象也不再存在。对象的表示被称为描述符,对象的描述符表示了须要知道的有关该对象的所有信息,见 6.3.4“描述符”。

3.3 提示性内容

在 GB/T 12991 中的某些地方,出现了提示性注解。例如:

注 2: 这是一个注解的例子。

这些注解并不属于 GB/T 12991 的标准内容,由于对非标准内容声明其符合性是毫无意义的,因此不要对注解内容声明符合性。

4 概念

4.1 防止误解的说明

本章中描述的概念,绝大多数在 GB/T 12991 其他部分中有准确的详细说明。如有出入,以其他部分的说明为准。

4.2 SQL 环境及其组成

4.2.1 SQL 环境

一个 SQL 环境包括：

——一个 SQL 代理。

——一个 SQL 实现。

——零个或多个 SQL 客户模块，包含适用于 SQL 代理的外部调用过程。

——零个或多个授权标识符。

——零个或多个用户映射。

——零个或多个例程映射。

——零个或多个目录，每个目录包含一个或多个 SQL 模式。

——包含模式内容所描述的 SQL 数据的场所，主要是基表。这种数据可以被看作“数据库”，但是 GB/T 12991 中并不使用“数据库”这个术语，因为这个词在一般的上下文中有不同的含义。

4.2.2 SQL 代理

SQL 代理就是引起 SQL 语句执行的主体。在 SQL 直接调用的情况下（见 5.3.3.3“SQL 直接调用”），它是由实现定义的。换句话说，SQL 代理可以由一个或多个编译单元组成。这些编译单元执行时调用 SQL 客户模块中的外部调用过程。

4.2.3 SQL 实现

SQL 实现是执行 SQL 代理所请求的 SQL 语句的处理器。一个 SQL 实现，由 SQL 代理可以看出，包括一个与 SQL 代理相关联的 SQL 客户和一个或多个 SQL 服务器。即使 SQL 环境中不允许多个 SQL 服务器存在，一个 SQL 实现也是可能符合 GB/T 12991 的。

由于 SQL 实现是依据其如何执行 SQL 语句规定的，所以这个概念通常表示某个安装的软件实例（数据库管理系统）。GB/T 12991 不区别软件供应商确定的和安装者确定的 SQL 实现之间的特征。

GB/T 12991 认为 SQL 客户和 SQL 服务器可能来自于不同的供应商，因此没有规定 SQL 客户与 SQL 服务器之间的通信方法。

4.2.3.1 SQL 客户

SQL 客户是一个处理器，SQL 代理将其视为 SQL 实现的一部分，它建立自身与 SQL 服务器之间的 SQL 连接，维护其自身、SQL 代理、SQL 服务器三者之间交互的诊断区和其他状态数据。

当为一个 SQL 客户建立了一个或多个 SQL 连接时，该 SQL 客户所维护的诊断区为当前 SQL 连接的 SQL 服务器所维护的诊断区栈（见 4.8.4）中第一个诊断区的副本。当不存在 SQL 连接时，该诊断区要么是空的，要么包含与某个失败的 SQL 连接相关的诊断信息。

4.2.3.2 SQL 服务器

每个 SQL 服务器是一个处理器，SQL 代理将其视为 SQL 实现的一部分，它管理 SQL 数据。

每个 SQL 服务器：

——管理发生在 SQL 服务器与 SQL 客户之间的 SQL 连接上的 SQL 会话；

——执行从 SQL 客户接收到的 SQL 语句，接收和发送要求的数据；

——维护 SQL 会话的状态，包括授权标识符和一些会话的缺省信息。

4.2.4 SQL 客户模块

一个 SQL 客户模块是由实现定义的机制显式创建和撤销的模块。

一个 SQL 客户模块的名字不是必须的；如果有，其允许的名字是由实现定义的。

一个 SQL 客户模块包含零个或多个外部调用过程。一个外部调用过程是由一条 SQL 语句组成的。外部调用过程由宿主语言的编译单元调用。

任何时候，只有一个 SQL 客户模块与一个 SQL 代理相关联。然而，在直接绑定方式或 SQL/CLI 方式下，SQL 客户模块可以是一个缺省的 SQL 客户模块，其存在对于用户是不可见的。

SQL 会话模块是为准备的 SQL 语句隐式创建的模块(见 GB/T 12991.2 中的 4.37“SQL 会话”)。

与一个 SQL 会话(而不是一个 SQL 会话模块)相关联的每个〈SQL 客户模块定义〉M 可能与一个影子模块 M1 相关联,在相同的 SQL 会话中,该影子模块具有一个实现相关的名字,该名字不等于任何其他〈SQL 客户模块定义〉的〈模块名〉,且其〈模块授权子句〉指明“SCHEMA SN”,其中 SN 是 M 的〈模块授权子句〉的显式或隐式〈模式名〉。M1 的〈语言子句〉、SQL 路径(若规定)和〈模块字符集说明〉等同于 M 所对应的特征。当 M 包含一个〈模块授权子句〉指明“FOR STATIC ONLY”,则该影子模块实际上包含 M 中所含的〈准备语句〉或〈立即执行语句〉所准备的每条 SQL 语句的一个〈外部调用过程〉。

4.2.5 用户标识符

用户标识符代表一个用户。创建和撤销用户标识符的方法,以及到真实用户的映射,在 GB/T 12991 中没有做出规定。

4.2.5a 角色

角色是特权或其他角色的潜在的接受者和授予者。角色也可以拥有模式和其他对象。

角色授权允许接受者(见 4.6.12)使用已授予该角色的所有特权。它也指明了该角色授权是否是可授予的,在这种情况下,接受者被授权可以将该角色授权、回收对该角色的授权及销毁该角色。

4.2.6 用户映射概念

一个用户映射把一个授权标识符与一个外部服务器描述符对应起来。

4.2.7 例程映射概念

一个例程映射把一个 SQL 调用例程与一个外部服务器描述符对应起来。

注 3: 按 GB/T 12991 的本版本,与外部服务器描述符对应的 SQL 调用例程被限制为 SQL 调用规则函数,将来的版本可能撤销这一限制。

4.2.8 目录和模式

4.2.8.1 目录

目录是在一个 SQL 环境中的 SQL 模式、外部服务器描述符、外部数据封装器描述符的命名集合。创建和撤销目录的机制是由实现定义的。

在当前 SQL 会话中,通过〈准备语句〉和〈立即执行语句〉的执行而动态准备的〈可准备语句〉的缺省目录名起初是由实现定义的,但可利用〈设置目录语句〉来改变。

4.2.8.2 SQL 模式

SQL 模式,通常简称为模式,是一个持久的、描述符的命名集合。任何其描述符属于某个 SQL 模式的对象被称为 SQL 模式对象。

与模式关联的授权标识符拥有该模式、模式中的模式对象以及由模式对象描述的 SQL 数据。

SQL 模式的创建和撤销通过执行 SQL 模式语句(或由实现定义的机制)来完成。

4.2.8.3 信息模式

每个目录均包含一个名为 INFORMATION_SCHEMA 的 SQL 模式,该模式包括一定数量的模式对象的描述符,大多数是视图定义,它们使得该目录中的每个描述符都能够像 SQL 数据一样被访问,但不能被修改。

通过信息模式中的视图可获得的数据包括该信息模式自身的描述符,不包括定义模式(见 4.2.8.4)中的模式对象和基表。

每个信息模式视图规定一个给定用户只能访问视图中该用户在其表示的描述符上具有特权的那些行。

4.2.8.4 定义模式

定义模式是一个虚构的名为 DEFINITION_SCHEMA 的模式。如果它真的存在,那么,其基表中的 SQL 数据应该描述一个 SQL 服务器可以获得的所有 SQL 模式。GB/T 12991 中的定义模式仅作为信息模式中视图的基础而定义。

定义模式的结构是 SQL 数据模型的一种表示。

4.2.9 外部服务器和描述符

外部服务器是一个处理器,它并不是 SQL 实现的一部分。一个外部服务器由一个外部服务器描述符来描述。外部服务器管理不属于 SQL 环境的数据。SQL 服务器和 SQL 客户能够使用外部服务器描述符(它是一个目录元素)与一个外部服务器通信。外部服务器管理的数据可由 SQL 服务器或 SQL 客户通过外部表(属于 SQL 模式元素)来访问。

4.2.10 外部数据封装器和描述符

外部数据封装器提供一种机制,SQL 服务器通过这种机制能够访问由外部服务器管理的数据。外部数据封装器由外部数据封装器描述符来描述。

4.2.11 SQL 数据

SQL 数据是由 SQL 模式描述的数据,即 SQL 环境中在 SQL 实现的控制之下的数据。

4.3 表

表具有一个或多个列的有序汇集和零个或多个行的无序汇集。每一列有一个名称和一个数据类型。每一行,相对于每一列而言,只有一个属于该列的数据类型的值。

完全由表变量组成的 SQL 数据称为基表。引用零个或多个基表且返回一个表的操作称为查询。查询的结果称为导出表。

表的行具有一个类型,称为“行类型”,表中的每一行具有相同的被称为“行类型”的类型。基于某个结构类型声明的表称为“类型表”,其列名和声明的类型对应于该结构类型中属性的名字和类型。类型表还有一个附加列,称为“自引用列”,其类型是与该表的结构类型相关联的引用类型。

如果类型表 TB1 有一个关联结构类型 TP1,TP1 是某结构类型 TP2 的子类型,则可以把 TB1 定义成其关联类型为 TP2 的类型表 TB2 的子表。这种情况下,TB2 称为 TB1 的“超表”。

视图是一个命名查询,能够通过其名字进行调用,这种的调用结果称为视图表。

有些查询,即有些视图是可更新的,这意味着它们可以作为改变 SQL 数据的那些语句的操作目标。视图上的更新最终还是体现为相应基表上的更新。

基表或视图表中,任意两列不能有相同的名字。除视图表外的导出表可以包含同名列。

基表可以是一个模式对象(其描述符属于模式,见 4.6.6),也可以是一个模块对象(其描述符属于模块,见 4.9)。描述符属于模式的基表称为创建基表,它可以是持久的或临时的(虽然两种情况的描述符都是持久的)。一个持久基表包括零或多行持久的 SQL 数据。模块中声明的基表只能是临时的,且称为声明临时表。

临时表是一个 SQL 会话对象,不能在其他 SQL 会话中进行访问。全局临时表可以在任何相关的 SQL 客户模块中访问。局部临时表只能在其所属的模块中访问。

当 SQL 会话初始时,临时表是空的。依据其描述符的描述,临时表会在 SQL 事务终止时,或是 SQL 会话终止时被清空(即所有的行都被删除)。

4.4 SQL 数据类型

4.4.1 一般数据类型信息

每个数据值都属于某个数据类型。

每个数据类型可以是预定义类型、构造类型或用户定义类型。每个数据类型有一个名字。预定义和构造数据类型的名字使用 GB/T 12991 中用来指定数据类型的保留字。用户定义类型的名字在其定义中指定。一个用户定义类型是一个模式对象,见 4.6.4。

一个预定义数据类型是 GB/T 12991 中指定的一种数据类型,因而它由 SQL 实现提供。虽然定义预定义数据类型时要求(或允许)用户提供相关参数(例如一个数的精度),但它依然是预定义的。

一个预定义数据类型是原子的。原子类型是指其值不能由其他数据类型的值构成的数据类型。虽然存在某些操作(SUBSTRING、EXTRACT)可以从一个字符串或一个日期时间值中提取一部分,但这

并不表示字符串类型或日期时间类型不是原子的。

一个构造类型可以是原子的,也可以是复合的。复合类型指的是这样一种数据类型,它的每个值由零个或多个值组合而成,其中每个值都属于一个已声明数据类型。

4.4.2 空值

每种数据类型都包含一个特殊值,称为空值,有时用关键词 NULL 表示。这个值在以下方面与其他的值不同:

- 由于每种数据类型都有空值,所以不能从关键字 NULL 表示的空值推断出其数据类型,因此 NULL 只能在某些特定的上下文中用来表示空值,而不是任何允许有字面出现的地方。
- 尽管空值既不等于任何其他值,也不是不等于任何其他值—它是否等于一个给定值是未知的。但在某些上下文环境中,多个空值可以一起处理,例如,〈GROUP BY 子句〉可以一起处理所有的空值。

4.4.3 预定义类型

4.4.3.1 数值类型

数值类型有两类:精确数值类型和近似数值类型。精确数值类型包括整数类型和具有指定精度和标度的类型;近似数值类型实质上是浮点数值类型,其精度说明是可选的。

每个数具有一个精度(数字的个数),精确数值类型还具有一个标度(小数点后的数字个数)。算术运算的操作数可以是相同或不同的数值类型,其运算结果的数值类型仅取决于操作数的数值类型。如果结果类型不能准确表示结果值,那么是进行四舍五入、还是截断,这是由实现定义的。如果结果值超出结果数据类型的表示范围,或是操作数上没有定义该算术运算,则将产生一个异常状态。

4.4.3.2 字符串类型

一个字符串类型的值是从某个字符表中抽出的一个字符串(序列)。一个字符串 S 中的字符都来自于同一个字符集 CS。如果 S 是某个表达式 E 的值,则 CS 是 E 的声明类型所指定的字符集。字符串可以是定长的,也可以是变长的(直到某个实现定义的上限)。一个字符大对象类型的值是来自于某个字符表的字符串,且总是只与一个字符集关联。一个大对象字符串是变长的,直到某个实现定义的上限,这个上限可能要比其他字符串的上限大。

字符串或字符大对象的字符集可以使用 CHARACTER SET 在数据类型中进行指定。如果实现选择的特定字符集是国家标准字符集,则可以使用 NATIONAL CHARACTER、NATIONAL CHARACTER VARYING、NATIONAL CHARACTER LARGE OBJECT(或其他语法上等价的方式)来说明数据类型。

一个二进制串类型(称为二进制大对象或 BLOB)的值是一个八位位组的变长序列,直到某个实现定义的上限。

4.4.3.3 布尔类型

一个布尔数据类型的值可以是真(TRUE)或假(FALSE)。未知(UNKNOWN)的值有时由空值来表示。

4.4.3.4 日期时间类型

日期时间类型有三种,每种类型的值由日期时间字段构成。

TIMESTAMP 数据类型的值由日期时间字段值 YEAR(从 0001~9999)、MONTH、DAY、HOUR、MINUTE 和 SECOND 组成。

TIME 数据类型的值由日期时间字段值 HOUR、MINUTE 和 SECOND 组成。

DATE 数据类型的值由日期时间字段值 YEAR(从 0001~9999)、MONTH 和 DAY 组成。

一个 DATE 的值是一个有效的格林尼治日期。一个 TIME 的值是一天中的有效时间。

TIMESTAMP 和 TIME 可以说明十进制的小数秒的精度。

TIMESTAMP 和 TIME 说明可以包含选项 WITH TIME ZONE,在这种情况下,每个值都与一个

时区偏移量相关联。在比较数据类型带有 WITH TIME ZONE 的时间值时,将会忽略其时区偏移量。

4.4.3.5 时间间隔类型

一个时间间隔类型的值表示一段持续的时间。时间间隔类型有两类:一类称为年月时间间隔,其时间间隔精度或包含 YEAR 字段,或包含 MONTH 字段,或两者都包含;另一类称为日时时间间隔,其指明或隐含的时间间隔精度可包含除 YEAR 字段和 MONTH 字段外的任何相邻字段的组合。

4.4.3.6 XML 类型

XML 类型的值称为 XML 值。

4.4.4 构造原子类型

4.4.4.1 引用类型

引用类型是一种构造数据类型,引用类型的值引用(或指向)某个拥有被引用类型的一个值的场所。能够这样被引用的场所只有类型表中的行,由此可见每个被引用类型必然是一个结构类型。

4.4.5 构造复合类型

4.4.5.1 集合类型

集合由零个或多个指定数据类型(称为元素类型)的元素构成。

数组是一个其值未必相异的有序集,其元素可通过它们在数组中的位置进行引用。

数组类型由数组类型构造符说明。

多重集是一个其值未必相异的无序集。

多重集类型由多重集类型构造符说明。

4.4.5.2 行类型

行类型是一个或多个被称为字段的(字段名,数据类型)对的序列。一个行类型的值由它的每个字段的一个值构成。

4.4.5.3 字段

字段是一个(字段名,数据类型)对。字段值是其数据类型的一个值。

4.5 场所及场所上的操作

4.5.1 场所

场所是一个保存指定数据类型值的实例的地方。每个场所具有一个定义的持久度,独立于其数据类型。除非有意撤销,否则一直存在的场所,称为持久的;在复合 SQL 语句、SQL 事务或 SQL 会话结束后就必须终止其存在的场所,称为临时的;仅用于保存一个变量或一个返回值的场所,则称为过渡的。

如上所述,主要的持久性场所或临时性场所是基表。基表是一种特殊类型的场所,在基表中可以指定其值的约束,并要求 SQL 实现强制执行这些约束(见 4.6.6.3)。

某些场所可以由其名称来引用,例如基表和 SQL 变量(见 GB/T 12991.4)。某些场所可以由 REF 值来引用。数组元素占用的场所可以由元素号来引用。

4.5.2 赋值

场所上的实例可以通过赋值操作来改变。赋值操作将场所上的实例(称为目标)替换为一个新的(可能不同的)值的实例(称为源)。赋值操作不影响对场所值的引用(如果有的话)。

4.5.3 可空性

每个场所都具有一个可空特性,表明该场所是可以包含空值(可能为空)还是不可以包含空值(已知不可空)。只有基表的列可以被约束为已知不可空,但由这些列导出的列也可以继承该特性。

虽然基表可以不包含任何行,但基表本身是不可为空的。

4.6 SQL 模式对象

4.6.1 一般 SQL 模式对象信息

每个 SQL 模式对象都有一个描述符。持久基表的描述符描述了一个与其分离的(尽管依赖于它)作为 SQL 数据而存在的持久对象。而其他描述符所描述的对象与其描述符在存在上则是不可区别的。

例如,当严格说来使用术语“断言描述符”更正确时,使用术语“断言”并不会影响表达的精确性。

每个模式对象都具有一个名字,这个名字在模式内的其所属的名称类对象中是唯一的。模式名称类对象包括:

- 基表和视图;
- 域和用户定义类型;
- 表约束,域约束和断言;
- SQL 服务器模块;
- 触发器;
- SQL 调用例程(仅指专用名,它不要求被显式指定,但如果没有显式指定,则是实现相关的);
- 字符集;
- 理序;
- 变换;
- 序列生成器。

某些模式对象具有命名的组成元素,这些组成元素的名字在其所属的对象中要求是唯一的。因此,列是基表或视图的唯一命名组成元素;属性是结构类型的唯一命名组成元素;字段是行类型的唯一命名组成元素。

有些模式对象可能是由 SQL 实现提供的。对于这些模式对象,用户既不能创建也不能撤销。

4.6.2 与字符集相关的描述符

4.6.2.1 字符集

字符集包括一个命名的字符集合(字符表),该字符集合用来形成字符串类型的值,以及一个命名的字符编码形式。每个字符集都有一种缺省理序。SQL 实现提供的字符集是由其他标准定义的还是由 SQL 实现定义的,这些信息在信息模式中表示。

当字符完全属于某个 SQL 实现时,字符的编码方法和形成字符串的方法是实现相关的。当与宿主程序或 SQL 实现以外的其他实体交换字符时,该字符集还有一种编码(说明表示每个字符的比特位)和字符编码形式(说明形成字符串的方式)。

SQL 实现所支持的每个字符集的字符表均为 ISO/IEC 10646 所说明的通用字符集的字符表的某个子集。

ISO/IEC 10646 的 6.5 中说明的符号是 GB/T 12991 中字符和字符串的标准表示。

注 4: ISO/IEC 10646 为“私用”字符指派了编码范围。ISO/IEC 10646 的将来版本可能增加更多编码点,这些编码点要求 SQL 实现提供支持。

4.6.2.2 理序

理序是对一个特定的字符表中的字符串进行排序的命名操作。

声明具有字符数据类型的场所可以指定一个理序,这个理序被当作其数据类型的一部分。

每个理序都应该是 由一个国际标准(如 ISO/IEC 14651)或国家标准定义或者推导出来的,也可以是实现定义的理序。

4.6.2.3 变换

变换是一个命名操作,该操作将某个字符集的字符串映射成给定字符集(不必不同)的字符串。该操作由变换名标识的外部函数调用来完成。由于变换时是将整个字符串传递给这个函数并返回一个字符串,因此映射不必是从一个字符到一个字符,而可以从一个或多个字符的序列到一个或多个字符的另一个序列。

4.6.3 域及其组成

4.6.3.1 域

域是一个命名用户定义对象,在能够说明数据类型的地方,可将该对象说明为一个数据类型的替

换。域由一个数据类型,一个可能有的缺省选项,以及零个或多个(域)约束组成。

4.6.3.2 域约束

域约束应用于基于该域的每一列,其操作就像表约束对表上的每一列一样。

域约束只能应用于基于其关联的域的那些列。

域约束被应用于该域上的转换操作的任何结果值。

4.6.4 用户定义类型

4.6.4.1 结构类型

结构类型是一个命名的用户定义数据类型。结构类型的值由若干属性值组成,结构类型的每个属性有一个数据类型,由属性类型来说明,属性类型包括在该结构类型的描述符中。

属性值是被封装的,即它们不能由用户直接访问。只有通过调用一个称为观察函数的函数才能访问属性值,属性值就是这个函数的返回值。一个结构类型的实例还可以通过定位器来访问。

一个结构类型可以被定义为另一个结构类型(称为直接超类型)的子类型。子类型继承其直接超类型的每个属性,且可以具有自己的附加属性。子类型的表达式可以出现在其任何超类型表达式可以出现的任何地方(这个概念称为可替代性),此外,表达式的值可以是声明类型表达式的任何子类型的值。

基于结构类型,可以创建一个或多个基表。基于结构类型 ST 的基表可以是基于 ST 的超类型的基表的子表。

基于结构类型,可以创建一个或多个视图。基于结构类型 ST 的视图可以是基于 ST 的超类型的视图的子视图。

4.6.4.2 属性

属性是结构类型的命名元素,它具有一个数据类型和一个缺省值。

4.6.5 相异类型

相异类型是一种基于某种预定义类型的用户定义数据类型。相异类型的值通过它所基于的类型的值来表示。

一个相异类型的变元只能被传递给相同相异类型的参数。这样,就可以对由这种数据类型的变元调用的例程进行精确的控制。

4.6.6 基表及组成

4.6.6.1 基表

基表是一个保存表数据(见 4.3)的场所。所有 SQL 数据都保存在基表中。

如果一个基表建立在一个结构类型上,它可能是一个或多个其他基表的子表,这些基表称为它的超表。

4.6.6.2 列

列是表的命名组成。它有一个数据类型、一个缺省值和一个是否可空的特性。

4.6.6.3 表约束

表约束是与单个基表相关联的完整性约束。

表约束可以是唯一性约束、主键约束、引用约束或检验约束。

唯一性约束规定表中一个或多个列为唯一性列。当且仅当表中唯一性列上不存在相同的非空值时,唯一性约束才被满足。

主键约束是通过 PRIMARY KEY 指定的唯一性约束。当且仅当表中唯一性列上不存在相同的非空值,且规定列上不存在空值时,主键约束才被满足。

引用约束规定一个或多个列为引用列,且对应于某个基表(不必是不同的表)中的被引用列,该基表称为被引用表。这样的被引用列是被引用表中某个唯一性约束的唯一性列。对于引用表中的每一行,如果每个引用列的值都等于被引用表中某一行相对应的每个被引用列的值,则引用约束总是被满足的。如果出现空值,则引用约束满足与否决定于指定的空值处理方式(称为匹配类型)。

通过规定引用动作,可以确定在被引用表数据发生变化导致违反引用约束时,应如何对引用表进行相应的修改。

表检查约束规定了一个搜索条件。对于表中的任何行,如果搜索条件的结果为假(不包括未知),就违反了检查约束。

4.6.6.4 触发器

触发器虽然未被定义成基表的组成元素,但却是与单个基表相关联的对象。每个触发器指明一个触发事件、一个触发动作时间、一个或多个触发动作。

触发事件指明基表上的什么操作将会引起触发动作。触发事件可以是 INSERT、DELETE 或 UPDATE。

触发动作时间指明所要采取的触发动作是在触发事件前(BEFORE)还是事件后(AFTER)执行。

触发动作或者是一个 SQL 过程语句;或者是 BEGIN ATOMIC,后跟以〈分号〉结束的一个或多个〈SQL 过程语句〉,后跟 END 的形式。

4.6.7 视图定义

视图(严格地说应为视图定义)是一个命名查询。视图具有多种用途,可以像基表一样使用。视图的值就是该查询的结果。见 4.3。

4.6.8 断言

断言就是一个检查约束。如果搜索条件的结果为假(不包括未知),就违反了该约束。

4.6.9 SQL 服务器模块

SQL 服务器模块(在 GB/T 12991.4,SQL/PSM 中定义)是一个属于模式对象的模块。见 4.9。

4.6.10 模式例程

模式例程是一个 SQL 调用例程,该例程是一个模式对象。见 4.10。

4.6.11 序列生成器

序列生成器是一种生成连续精确数值的机制,每次生成一个。序列生成器或者是一个外部序列生成器,或者是一个内部序列生成器。外部序列生成器是一个命名模式对象,而内部序列生成器则是另一个模式对象的元素。序列生成器有一个数据类型,且应是标度为 0 的精确数值类型。序列生成器还有一个随时间变化的当前基值,该值为其数据类型的一个值。详见 GB/T 12991.2 中 4.21。

4.6.12 特权

特权代表某个授权者对权限的指定接受者(授权标识符、角色或 PUBLIC)的一种认可,即要求使用指定的模式对象,或在规定的模式对象上实施指定动作的权力。其具体动作包括 SELECT、INSERT、UPDATE、DELETE、REFERENCES、USAGE、UNDER、TRIGGER 和 EXECUTE。

带有 WITH GRANT OPTION 的特权允许接受者把该特权授予其他的接受者,接受者授权时可以带也可以不带 WITH GRANT OPTION 选项。

带有 WITH HIERARCHY OPTION 的 SELECT 特权自动提供给接受者被授权表的所有子表上的 SELECT 特权,包括现存的和将来会被加入的子表。

PUBLIC 代表可以接受特权的每个接受者。对信息模式中的所有模式对象,带 WITH GRANT OPTION 的 SELECT 特权已授予 PUBLIC。

创建模式对象的授权标识符自动被授予该对象上的所有可能的特权,且包括 WITH GRANT OPTION。

只有拥有模式对象上的某个特权的授权标识符才能够发现该对象的存在。

4.7 完整性约束和约束检查

4.7.1 约束检查

有两种模式对象描述约束:断言和表约束(包括表中某一列所基于的域的域约束),它们用相同的方法来检查。

每种约束或者是可延迟的,或者是不可延迟的。

在每个 SQL 会话中,每个约束都具有一个约束模式,它是该 SQL 会话的一个性质。每个约束都有一个(持久的)缺省约束模式,在每个 SQL 会话中的每个 SQL 事务,约束以缺省约束模式开始。

约束模式或者是延迟的,或者是立即的,只要约束是可延迟的,就可以使用 SQL 语句进行设置。

当事务初始化时,每个约束的约束模式被设置为其缺省方式。

如果约束模式是立即的,则当每条 SQL 语句执行结束时,对每个约束都会进行约束检查。

在事务终止前,每个约束的约束模式都置为立即的(所以这时将检查所有的约束)。

4.7.2 确定性和约束

表达式计算结果可能是非确定性的。例如,在列的数据类型为变长字符串的情况下,即使数据是相同的,由于时机不同,消除重复值后的剩余数据也可能不同。出现这种情况是因为重复值尾部的空格数可能不同,而 GB/T 12991 并没有规定消除重复值时应该保留其中的哪个值。因此,值的长度是不确定的。在这样的情况下,该表达式及从该表达式导出的任何表达式被称为是可能非确定性的(所谓“可能”,是因为曾经更新该列的部分 SQL 代理可能删除了尾部空格,但 SQL 实现是不可能知道这一情况的)。

由于包含一个可能非确定性的表达式的约束可能会一时满足条件,稍后又变得不满足,因此,不允许约束包含非确定性表达式。

例程可以声明为确定的,否则调用例程的结果是实现相关的。

4.8 SQL 代理和 SQL 服务器之间的通信

4.8.1 宿主语言

只有当 SQL 代理使用的某种编程语言符合 GB/T 12991 规定的标准时,SQL 实现与 SQL 代理才能成功地进行通信。这种语言通常称为宿主语言,符合 SQL 实现至少要支持一种宿主语言。

有几种称为绑定方式的通信方法:

——SQL 客户模块绑定方式(在 GB/T 12991.2 中说明)。在这种绑定方式中,用户通过使用实现定义的一种机制,指定一个模块作为 SQL 客户模块。

——调用层接口(在 GB/T 12991.3 中说明)。在这种方式中,SQL 代理在标准例程中选择一个进行调用,传递适当的变元,例如内容为某 SQL 语句的字符串。

——嵌入式 SQL(在 GB/T 12991.2 中说明)。在这种方式中,SQL 语句直接编写在应用程序中。此外还有一套实现相关的机制,用于:

- 为每条 SQL 语句生成一个外部调用过程。这些过程将被收集到一个模块中,以后作为 SQL 客户模块使用。
- 将每条 SQL 语句替换为对其生成的外部调用过程的调用。

——直接 SQL 调用(在 GB/T 12991.2 中说明)。直接调用是一种直接执行 SQL 语句的方法,通过前端直接和用户进行通信。

无论选择哪种绑定方式,SQL 语句都是按照实现定义的字符集(称为源语言字符集)来编写的。并不要求源语言字符集与出现在 SQL 数据中任何字符串的字符集相同。

GB/T 12991.2 规定了当 SQL 客户模块中的外部调用过程被符合标准要求的宿主语言程序调用时,外部调用过程的动作。

4.8.2 参数传递和数据类型对应性

4.8.2.1 一般参数传递和数据类型对应性信息

外部调用过程参数表中的每一个参数都有一个名字和一个数据类型。在外部例程对编译单元的引用和该编译单元之间进行绑定的时间和方法是实现定义的。

4.8.2.2 数据类型对应性

GB/T 12991 规定了 SQL 数据类型和宿主语言数据类型之间的对应关系。并不是每一种 SQL 数

据类型在每一种宿主语言中都有一个对应的数据类型。

4.8.2.3 定位器

宿主变量、宿主参数、外部例程的 SQL 参数、外部函数的返回值都可以规定为定位器。设置定位器的目的是：当要操作大规模数据实例时，不需向 SQL 代理来回转移整个数据。详见 GB/T 12991.2 中的 4.29.5。

4.8.2.4 状态参数

每个外部调用过程都要求有一个输出参数 SQLSTATE，称为状态参数。

SQLSTATE 是一个长度为 5 的字符串，具体取值在 GB/T 12991 其他部分中定义。值为“00000”（五个零）的 SQLSTATE 表示，最近一次的外部调用过程的调用是成功的。

4.8.2.5 指示符参数

指示符参数是一个整型参数，紧随其相关联的参数（而不是指示符参数）后。值为负的指示符参数表示其相关的参数为空；指示符参数的值大于零表示其相关参数值的长度需要截断，该指示变量存放了该列值的实际长度，通常在字符串及某些其他数据类型的情况下会出现。

如果一个参数的值被赋为空值，且没有相关的指示符参数，则会产生一个异常状态。

4.8.3 描述符区

描述符区是一个由 SQL 实现在 SQL 代理的要求下分配的命名区。描述符区用于 SQL 实现和 SQL 代理之间的通信，有些 SQL 语句可在 SQL 代理和描述符区之间传递信息。

4.8.4 诊断信息

诊断区是一个由 SQL 实现分配的通信区，能够保存一个或多个发生的状态。

当 SQL 实现执行一条 SQL 语句（SQL 诊断语句除外）时，就在诊断区中设置一个或多个状态信息的值，以表示执行语句所发生的情况。通过使用 SQL 诊断语句，可以访问诊断信息。诊断语句的执行信息不被记录。

符合 SQL 实现不要求同时设置多于一个的状态信息。

当 SQL 服务器执行一条 SQL 语句 SS 时，可能存在多个诊断区。当 SS 执行完成之前，由于 SS 的执行引起了另外的 SQL 语句同时执行时，就会发生这种情况。多诊断区形成栈，栈的第一个元素包含最近所执行的 SQL 语句相关的信息。这种情况的两个例子是：SS 是或者包含一个（调用语句）；SS 的执行引发了触发动作的执行。

4.8.5 SQL 事务

SQL 事务是一个 SQL 语句的执行序列，对于恢复它是原子的。也就是说，执行结果要么是完全成功的，要么对于任何 SQL 模式或 SQL 数据都不起作用。

任何时刻，在 SQL 代理和 SQL 实现之间最多只能有一个当前 SQL 事务。

如果没有当前 SQL 事务，事务初始化语句的执行或（子查询）的计算将会初始化一个 SQL 事务。SQL 事务由提交语句或回滚语句终止，这两种语句的执行可能是隐式的。

SQL 事务具有一个事务状态。事务状态的某些性质通过 SQL 语句的执行来设置。只有不存在当前 SQL 事务时，这样的 SQL 语句才会被执行。事务初始化时或初始化之后，当 SQL 客户与 SQL 服务器连接或设置连接时，这些性质将在第一时间传递给 SQL 服务器。

SQL 事务的访问方式表示该事务是只读的（不允许改变任何持久 SQL 数据）还是可读写的（允许改变持久 SQL 数据）。

SQL 事务的隔离级规定在事务内感受到的 SQL 环境之外的 SQL 代理的行为效果的程度。

每个隔离级都要保证每个 SQL 事务被执行，SQL 事务不是完全执行完就是完全失败。每个隔离级都要保证 SQL 事务的更新不会丢失。最高隔离级可串行化（SERIALIZABLE）保证执行的可串行化，这意味着在时间上重叠的多个事务执行的效果与它们依次执行的效果完全相同。其他的隔离级包括可重复读（REPEATABLE READ）、读提交（READ COMMITTED）、读未提交（READ UNCOM-

MITTED)保证的隔离级是依次降低的。

4.9 模块

有三种不同类型的模块,每种模块都具有一些具体的性质,且包含不同种类的模块对象(也称为模块内容)。主要模块对象是一个或多个例程(见 4.10)。

模块是下列描述之一:

- 一个 SQL 客户模块,仅包含外部调用过程;
- 一个 SQL 服务器模块,仅包含 SQL 调用例程;
- 一个 SQL 会话模块,仅包含该 SQL 会话中已准备的 SQL 语句。

4.10 例程

4.10.1 一般例程信息

表 1 表示了用 SQL 或其他语言编写的例程及例程在 SQL 或其他语言中被调用的各种可能的组合情况,并给出了在这些情况下使用的术语。

表 1 外部调用例程和 SQL 调用例程间的关系

例程类别	用 SQL 编写的例程	非 SQL 编写的例程
由 SQL 调用的例程	SQL 函数和 SQL 过程	外部函数和过程
由非 SQL 调用的程	外部调用过程	(与 SQL 无关)

SQL 调用例程是一个能够在 SQL 中调用的例程,它或者是一个函数或者是一个过程。有些函数具有一些特定的性质,这种函数被称为方法。

SQL 调用例程可以是一个模式对象也可以是一个 SQL 服务器模块(本身为模式对象)的成员。

SQL 调用过程是一个由 SQL 调用语句(CALL 语句)调用的过程。SQL 调用函数由某个表达式中的例程调用来调用。

外部例程指的是引用 SQL 环境之外的特定标准程序设计语言的某个编译单元的 SQL 调用例程。这种引用的绑定方法和时间是由实现定义的。

SQL 例程是指例程体是用 SQL 编写的 SQL 调用例程。

SQL 调用例程的名字不必唯一。如果两个或更多的例程具有相同的名字,那么该名字被称为重载的,对同名例程的调用将调用变元最佳匹配的例程。正常情况下,在进行变元类型最佳匹配的过程中,仅考虑表示变元值的表达式的声明类型,但是对于方法而言(其调用语法是有区别的),还必须考虑变元的最有效类型。

4.10.2 类型保留函数

如果 SQL 调用函数有一个参数说明为 RESULT,该参数称为结果参数,如果结果参数的数据类型和结果的数据类型为相同的用户定义类型,则该函数称为类型保留函数。类型保留函数的结果是结果参数的值,这个结果可能是经过变异的。

每个变异函数都是一个类型保留函数。

4.11 SQL 语句

4.11.1 SQL 语句分类

SQL 语句是符合 GB/T 12991 中规定的格式和句法规则的字符串。

大多数 SQL 语句都可以在 SQL 客户模块中进行准备,然后执行。在这种情况下,当准备 SQL 语句时就创建了一个对应于单条 SQL 语句的外部调用过程,且每当执行准备的 SQL 语句时就会隐含调用该外部调用过程。

SQL 语句分类的方法至少有五种:

- 根据它们对 SQL 对象的作用,是持久对象(即:SQL 数据、SQL 模式及其内容、SQL 客户模块)还是临时对象(例如 SQL 会话和其他 SQL 语句);

- 根据它们是否初始化了一个 SQL 事务,或者当没有活动的 SQL 事务时,该语句是否能够或需要被执行;
- 根据它们是否可以嵌入到用标准程序设计语言编写的程序中;
- 根据它们是否可以直接执行;
- 根据它们是否可以动态准备并执行。

GB/T 12991 允许实现提供附加的、实现定义的语句,这些语句也可以归入上述类别。

4.11.2 按功能分类的 SQL 语句

下面是 SQL 语句的主要分类:

- SQL 模式语句,用来创建、变更、撤销模式和模式对象;
- SQL 数据语句,用来对表执行查询、插入、更新、删除操作,一个 SQL 数据语句的执行可能影响到多个表中的多行记录;
- SQL 事务语句,用来设置事务参数、启动或结束事务;
- SQL 控制语句,可用来对一个 SQL 语句序列的执行进行控制;
- SQL 连接语句,用来初始化和终止连接,允许 SQL 客户从与一个 SQL 服务器的会话转移到与另一个 SQL 服务器的会话;
- SQL 会话语句,用来设置 SQL 会话的缺省值和其他参数;
- SQL 诊断语句,用来(从诊断区)获取诊断信息和报告 SQL 例程中的异常信号;
- 动态 SQL 语句,支持动态产生的 SQL 语句的准备和执行,并获取其相关信息。

5 GB/T 12991 中的各部分说明

5.1 概述

GB/T 12991.1,框架,是 GB/T12991 其他各部分的引导,该部分描述了其他各部分使用的基本概念和记法。

GB/T 12991.2,基础,规定 SQL 语句的结构和执行 SQL 语句的效果。

GB/T 12991 中除了第 1 部分和第 2 部分外的其他各部分都是作为 GB/T 12991 的补充来规定的。详见 6.3.5。

GB/T 12991.3,调用层接口,规定 SQL 代理和 SQL 实现之间的另一种通信机制。

GB/T 12991.4,持久存储模块,规定使得 SQL 计算功能更加完备的重要的附加部分。

GB/T 12991.9,外部数据管理,规定 SQL 重要的附加部分,即允许 SQL 代理访问不受 SQL 环境中 SQL 服务器控制的数据。

GB/T 12991.10,对象语言绑定,规定 SQL 语句嵌入到 Java 程序中的方式。

GB/T 12991.11,信息和定义模式,规定一些视图,应用通过这些视图能够获悉持久数据库对象(如表、视图、列)的名字。

GB/T 12991.13,使用 Java 程序设计语言的 SQL 例程和类型,规定将 Java 语言编写的静态方法当作 SQL 调用例程来调用,以及将 Java 语言中定义的类型作为 SQL 结构类型来使用的功能。

GB/T 12991.14,与 XML 相关的规范,定义数据库语言 SQL 与 XML 结合使用的方法。

每一部分的内容将在下面分别介绍。

5.2 GB/T 12991.1:框架

GB/T 12991 本部分包括:

- SQL 环境的描述,以及 GB/T 12991 中使用的一些概念的简要描述;
- 每部分内容的简要描述,这些描述是资料性的,不是标准的组成部分;
- 适用于 GB/T 12991 中所有或大多数部分的记法和约定,其他各部分根据需要对约定作了进一步说明。

5.3 GB/T 12991.2:基础

GB/T 12991.2 规定 SQL 的以下特征。

5.3.1 GB/T 12991.2 规定的数据类型

以下是 GB/T 12991.2 中所规定的数据类型：

- 所有的数值类型和字符串类型；
- 布尔类型；
- 所有的日期时间和时间间隔类型；
- 行类型；
- 集合类型；
- 用户定义类型；
- 域；
- 引用类型。

5.3.2 表

定义确定表的函数依赖和候选键的规则。

5.3.3 绑定方法

GB/T 12991.2 规定了绑定 SQL 代理与 SQL 实现的三种方法。

5.3.3.1 嵌入式 SQL

嵌入式 SQL 是一种将 SQL 语句嵌入到符合标准指定的特定程序语言的编译单元中的方法，这种程序设计语言称为宿主语言。它定义了如何在宿主语言的环境中，按照特定宿主语言的规范，导出一个等价的编译单元。在该等价编译单元中，每条嵌入的 SQL 语句由调用包含该 SQL 语句的数据库语言过程的一条或多条语句所替代。

5.3.3.2 动态 SQL

动态 SQL 是嵌入式 SQL 的扩展。这些设施的作用如下：

- 分配和释放用于 SQL 代理和 SQL 实现之间通信的描述符区；
- 促成 SQL 语句的执行，包括为随后执行的语句作准备。

5.3.3.3 SQL 直接调用

SQL 直接调用是一种直接执行 SQL 语句的方法。在 SQL 直接调用过程中，以下内容是由实现定义的：

- 调用 SQL 语句的方法；
- 报告因执行这种 SQL 语句而产生的状态的方法；
- 访问因执行这种 SQL 语句而产生的诊断信息的方法；
- 返回结果的方法。

5.3.4 GB/T 12991.2 中规定的 SQL 语句

以下是 GB/T 12991.2 中规定的几类 SQL 语句：

- SQL 模式语句，用来创建、变更、撤销 GB/T 12991.2 中规定的模式和模式对象；
- SQL 数据语句，用来执行对表的查询、插入、更新、删除操作，有些 SQL 数据语句在其名字中包含词“dynamic”，不要与 SQL 动态语句混淆；
- SQL 事务语句，用来设置事务性质、初始化或结束事务；
- SQL 控制语句(CALL 和 RETURN)，分别用来调用一个过程及说明函数的返回值；
- SQL 连接语句，用来初始化和终止连接，允许 SQL 客户从与一个 SQL 服务器的会话转移到与另一个 SQL 服务器的会话；
- SQL 会话语句，用来设置 SQL 会话的缺省值和其他参数；
- SQL 诊断语句，用来(从诊断区)获取诊断信息；

- SQL 动态语句,支持动态生成的 SQL 语句的准备和执行,并获取这些语句的有关信息;
- SQL 嵌入异常声明,这些声明被转换成宿主语言中的语句。

对于定义的每条 SQL 语句,GB/T 12991.2 规定了在当前没有活动事务的情况下,哪些语句会初始化一个新事务,哪些语句不会。

对于每条 SQL 语句,GB/T 12991.2 规定了:

- 是否可以嵌入到宿主语言中。
- 是否可以动态地准备和执行。除了那些要获取数据到描述符区中的语句外,任何可准备的 SQL 语句都可以立即执行。
- 是否可以直接执行。

5.4 GB/T 12991.3:调用层接口

GB/T 12991.3 规定用标准程序设计语言编写的应用程序和 SQL 实现之间的绑定方法。其效果在功能上等价于 GB/T 12991.2(SQL/基础)中说明的动态 SQL。

规定的过程(例程)能够用来:

- 分配和释放资源(包括描述符或通信区);
- 初始化、控制和终止 SQL 客户和 SQL 服务器之间的 SQL 连接;
- 促成 SQL 语句的执行,包括为随后执行的语句作准备;
- 获取诊断信息;
- 获取有关 SQL 实现的信息,例如,SQL 客户能够连接的 SQL 服务器。

调用层接口 CLI 和绑定的重要区别在于:在后者的上下文中仅有一个 SQL 环境;而对于调用层接口而言,能够独立地初始化和多个 SQL 环境。所以,尽管 SQL 环境被简单地定义为具有各种特征的环境集合,但在 CLI 中,该术语指的是一个 SQL 环境的当前状态(描述符),这个 SQL 环境可能是多个 SQL 环境中的一个。因此,该术语用来表明应用(SQL 代理)和 SQL 客户之间的会话。而 CLI 中的 SQL 会话指的是 SQL 客户和 SQL 服务器之间的 SQL 连接。

5.5 GB/T 12991.4:持久存储模块

GB/T 12991.4 规定了附加 SQL 语句的句法和语义,使得 SQL 计算功能更加完备。

这些语句包括如下功能:

- 用于控制流程的语句说明;
- 将表达式的结果赋给变量和参数;
- 允许复合语句对其执行期间产生的各种状态进行处理的状态句柄说明;
- 设置信号状态和重置信号状态的语句说明;
- 局部游标声明;
- 局部变量声明。

本部分还定义信息模式表,这些信息模式表包含描述 SQL 服务器模块的模式信息。

5.5.1 GB/T 12991.4 中规定的 SQL 语句

下面是 GB/T 12991.4 中规定的 SQL 语句的大致分类及功能为:

- 附加的 SQL 控制语句,可用来控制 SQL 例程的执行;
- SQL 控制声明,可用来声明变量和异常句柄;
- 附加的 SQL 诊断语句,可用来设置异常信号;
- 附加的 SQL 模式语句,可用来创建和撤销模块。

5.6 GB/T 12991.9:外部数据管理

GB/T 12991.9 定义扩展的 SQL 数据库语言,以支持使用外部表和数据链接数据类型来进行外部

数据管理。

包括定义下列对象的机制：

- 外部服务器；
- 外部数据封装器；
- 外部表。

此外,还包括这样的例程,它们能够用于 SQL 服务器和外部数据封装器之间通信,从而实现和外部数据源之间的交互。

5.7 GB/T 12991. 10:对象语言绑定

GB/T 12991. 10 定义在 Java 程序中嵌入 SQL 语句的机制。

5.8 GB/T 12991. 11:信息和定义模式

GB/T 12991. 11 定义了信息模式 (INFORMATION_SCHEMA) 和定义模式 (DEFINITION_SCHEMA) 两个模式。应用程序可以通过信息模式提供的视图了解到持久数据库对象的名称,包括表、视图、列等。这些视图是根据定义模式中的基表进行定义的。定义模式的唯一目的就是提供一个支持信息模式的数据模型,并有助于理解。从信息模式视图的角度来看,SQL 实现仅需要模拟定义模式的存在。

5.9 GB/T 12991. 13:使用 Java 程序设计语言的 SQL 例程和类型

GB/T 12991. 13 定义对数据库语言 SQL 的扩展,使之能够将 Java 语言编写的静态方法当作 SQL 调用例程来调用,并且将 Java 语言中定义的类作为 SQL 结构类型来使用。

内容包括：

- 对 SQL 调用例程的定义、操纵和调用的扩展；
- 用户定义类型的定义和操纵的扩展；
- 新的内置过程。

5.10 GB/T 12991. 14:与 XML 相关的规范

GB/T 12991. 14 定义对数据库语言 SQL 的扩展,使之能够创建和操纵 XML 文档。

内容包括：

- 新的预定义类型 XML；
- 创建和操纵 XML 类型数据的新内置操作符；
- 表、模式和目录与 XML 文档之间的映射规则。

6 GB/T 12991 中其他各部分使用的记法和约定

本章中定义的记法和约定用于 GB/T 12991 中其他各部分,除非其他各部分提供有更确切的定义。

6.1 取自 ISO/IEC 10646 中的记法

表示 UCS 代码点的记法定义于 [UCS], 见 6.5“短字符标识符”。

在本标准中,该记法仅用来无歧义地标识字符,并不隐含任何实现使用该字符的特定编码。

6.2 本标准中使用的记法

GB/T 12991 中使用的句法记法是 BNF (“巴科斯范式”或“巴科斯诺尔范式”)的扩展版本。

在 BNF 语言定义中,语言的每一个句法元素(称为 BNF 非终结符)都由产生式规则来定义。它采用公式的形式定义句法元素,每个公式由一些字符、字符串及句法元素组成,该公式可用来生成句法元素的实例。

GB/T 12991 使用的 BNF 版本中,具体符号的含义如表 2 所示。

表 2 BNF 中使用的符号

符号	含 义
< >	尖括号内的字符串表示 SQL 语言的一个句法元素(BNF 非终结符)的名称
::=	定义符,用于产生式规则,用来分割规则定义的元素及其定义。被定义的元素出现在定义符的左边,而定义元素的公式出现在定义符的右边
[]	方括号表示公式中的可选元素。方括号内公式中的部分可以显式指定也可以省略
{ }	花括号括着公式中的分组元素。花括号内公式中的部分必须显式说明
	选择符。竖线表示在公式中后边的部分可以代替在它前面的部分。如果竖线不是出现在方括号或大括号中,则表示由产生式规则定义的元素可相互完全替换。如果竖线出现在方括号或大括号中,则表示竖线所在最内层大括号或方括号中的内容可互相替换
...	省略号表示公式中使用了该符号的元素可以重复任意多次。如果省略号直接出现在大括号“}”之后,则它应用于该大括号“}”和对应的大开括号“{”之间所括的公式部分。如果省略号出现在任意其他元素之后,则仅用于该元素。在“语法规则”、“访问规则”、“一般规则”和“符合性规则”中,如果没有特别说明,则对由此生成的列表中第 n 个元素的引用就表示列表中该序号对应的元素
!!	引入一段汉语正文。当语法元素没有用 BNF 表示时使用

空格用来分隔句法元素。多个空格和换行作为单个空格处理。除了上面给出的那些有特殊功能的符号外,公式中的其他字符和字符串均代表它们自身。另外,如果产生式中定义符右边的符号完全由 BNF 符号组成,则这些符号也代表它们自身,不再具有它们的特殊含义。

成对大括号和方括号对可以多重嵌套,选择符可在上述嵌套中的任意一层出现。

构成句法元素实例的字符串根据该句法元素的 BNF 定义生成,步骤如下:

- 1) 从产生式规则右边定义的选项中任选其中一个,并用这个选项代替该元素;
- 2) 将每个省略号及其作用的对象替换为一个或多个该对象的实例;
- 3) 对方括号所括字符串中的每一个部分,或者删除该括号及其内容,或者将方括号改为大括号;
- 4) 对每个大括号中所括的内容,应用步骤 1)到步骤 5)进行处理,然后删除大括号;
- 5) 对字符串中还存在的 BNF 非终结符,应用步骤 1)到步骤 5)进行处理。

当字符串中不再含有任何非终结符,展开或生成过程完成。

由 BNF 非终结符 NT 在原语言字符集中生成一个字符串 CS 的左范式推导,可以通过应用步骤 1)到步骤 5)完成,并且在步骤 5)中,总是先挑选最左边的 BNF 非终结符来处理。

6.3 约定

6.3.1 句法元素说明

句法元素规定如下:

- 功能:元素用途的简要描述。
- 格式:该元素句法的 BNF 句法定义。
- 句法规则:元素句法性质的说明,或 BNF 范式无法表示而该元素应该满足的附加的句法约束的规范,或是上述两者。
- 访问规则:对控制模式对象可存取性规则的说明,这些规则应该作用在一般规则之前。
- 一般规则:元素运行时效果的说明。当用多条一般规则来说明元素的效果时,由第一条一般规则开始,按数字顺序应用这些规则,将获得所需要的效果,除非所用规则指明或隐含的顺序有变化或规则应用的终止。除非特定规则另有所指或另有隐含,只有按顺序应用了最后一条规则之后,一般规则的应用才能终止。
- 符合性规则:关于应如何支持该元素以保证 SQL 符合性的说明。

记法的范围是定义这些符号的条款。在一个条款中,句法规则、访问规则、一般规则中所定义的符

号可以被其他规则引用,只要它们在被引用之前已定义。

6.3.2 信息模式和定义模式的说明

GB/T 12991 中信息模式和定义模式的对象规定如下:

- 功能:关于该定义的用途的简要说明。
- 定义:用 SQL 对该对象进行定义。
- 描述:对象运行时的值的说明,由于(对对象运行时的值)定义不清晰而增加该描述。
- 符合性规则:关于应如何支持该元素以保证 SQL 符合性的说明。

信息模式中视图定义的唯一作用就是指明那些视图表的内容。这些视图基于的实际对象是实现相关的。

6.3.3 术语的使用

6.3.3.1 句法含义

令 $\langle A \rangle$ 、 $\langle B \rangle$ 和 $\langle C \rangle$ 是句法元素;令 A_1 、 B_1 和 C_1 分别是 $\langle A \rangle$ 、 $\langle B \rangle$ 和 $\langle C \rangle$ 的实例。

在格式中,如果 $\langle B \rangle$ 出现在 $\langle A \rangle$ 的 BNF 产生式规则的右边,则称 $\langle A \rangle$ 立即包含 $\langle B \rangle$ 。如果 $\langle A \rangle$ 立即包含 $\langle C \rangle$,或 $\langle A \rangle$ 立即包含 $\langle B \rangle$,而 $\langle B \rangle$ 又包含 $\langle C \rangle$,则称 $\langle A \rangle$ 包含或规定 $\langle C \rangle$ 。

在 SQL 语言中,如果 $\langle A \rangle$ 立即包含 $\langle B \rangle$,且 B_1 是 A_1 的文本的一部分,则称 A_1 立即包含 B_1 。如果 A_1 立即包含 C_1 ,或 A_1 立即包含 B_1 ,而 B_1 包含 C_1 ,则称 A_1 包含或规定 C_1 。如果 A_1 包含 C_1 ,则 C_1 包含于 A_1 且 C_1 由 A_1 规定。

如果 A_1 包含 B_1 ,且 A_1 包含 $\langle C \rangle$ 的一个实例,而 $\langle C \rangle$ 又包含 B_1 ,则称 A_1 包含带中介 $\langle C \rangle$ 的 B_1 。如果 A_1 包含 B_1 ,且 A_1 不包含 $\langle C \rangle$ 的一个实例,而 $\langle C \rangle$ 又包含 B_1 ,则称 A_1 包含不带中介 $\langle C \rangle$ 的 B_1 。

如果 A_1 包含不带中介 $\langle A \rangle$ 或中介 $\langle B \rangle$ 的实例的 B_1 ,则称 A_1 简单包含 B_1 。

如果 A_1 包含不带中介 \langle 子查询 \rangle 、 \langle 查询多重集值构造符 \rangle 、 \langle 查询表值构造符 \rangle 、 \langle 查询数组值构造符 \rangle 、 \langle 内分组说明 \rangle 或 \langle 集函数说明 \rangle (不是 \langle 有序集函数 \rangle) 的 B_1 ,则称 A_1 直接包含 B_1 。

如果 $\langle A \rangle$ 包含 $\langle B \rangle$,则称 $\langle B \rangle$ 被包含在 $\langle A \rangle$ 中,且称 $\langle A \rangle$ 是包含 $\langle B \rangle$ 的产生式符号。如果 $\langle A \rangle$ 简单包含 $\langle B \rangle$,则称 $\langle B \rangle$ 被简单包含在 $\langle A \rangle$ 中,且称 $\langle A \rangle$ 是简单包含 $\langle B \rangle$ 的产生式符号。

如果 A_1 满足条件 C 且 A_1 不包含满足 C 的 $\langle A \rangle$ 的实例,则称 A_1 是满足条件 C 的最内层的 $\langle A \rangle$ 。如果 A_1 满足条件 C 且 A_1 不被包含在满足 C 的 $\langle A \rangle$ 的实例中,则称 A_1 是满足条件 C 的最外层的 $\langle A \rangle$ 。

如果 $\langle A \rangle$ 包含一个 \langle 表名 \rangle ,该 \langle 表名 \rangle 标识了由 \langle 视图定义 $\rangle V$ 定义的视图,则称 $\langle A \rangle$ 一般包含 V 中包含的 \langle 查询表达式 \rangle 。如果 $\langle A \rangle$ 包含一个 \langle 查询表达式 $\rangle QE$ 标识的 \langle 查询名 \rangle ,则称 $\langle A \rangle$ 一般包含 QE 。如果 $\langle A \rangle$ 包含一个 \langle 例程调用 $\rangle RI$,则称 $\langle A \rangle$ 一般包含 RI 的目标例程集中所有 \langle SQL 调用例程 \rangle 的例程体。如果 $\langle A \rangle$ 包含 $\langle B \rangle$,则 $\langle A \rangle$ 一般包含 $\langle B \rangle$ 。如果 $\langle A \rangle$ 一般包含 $\langle B \rangle$ 且 $\langle B \rangle$ 一般包含 $\langle C \rangle$,则 $\langle A \rangle$ 一般包含 $\langle C \rangle$ 。

注 5:“ \langle 例程调用 \rangle 的目标例程集”在 GB/T 12991.2 的 10.4“例程调用”中定义。

在格式中,动词“to be”(包括所有的语法变种,例如“is”)定义如下:如果存在形如 $\langle A \rangle ::= \langle B \rangle$ 的一个 BNF 产生式规则,则称 $\langle A \rangle$ 是 $\langle B \rangle$ 。如果 $\langle A \rangle$ 是 $\langle B \rangle$ 且 $\langle B \rangle$ 是 $\langle C \rangle$,则 $\langle A \rangle$ 是 $\langle C \rangle$ 。如果 $\langle A \rangle$ 是 $\langle C \rangle$,则称 $\langle C \rangle$ 构成 $\langle A \rangle$ 。在 SQL 语言中,如果 $\langle A \rangle$ 是 $\langle B \rangle$ 且 A_1 的正文是 B_1 的正文,则称 A_1 是 B_1 ;反之,如果 A_1 是 B_1 ,则称 B_1 构成 A_1 。

6.3.3.2 表示规则要求的术语

在句法规则中,术语应该定义句法上要求符合的 SQL 语言为真的条件。当这样的条件依赖于一个或多个模式的内容时,则在一般规则所指明的动作执行之前要求它们为真。对于不符合 SQL 格式和句法规则的语言的处理是实现相关的。当试图运用访问规则或一般规则求值时,如果有任何句法规则要求的条件未被满足,且实现既不是在处理不符合的 SQL 语言,也不是在以不符合的方式处理符合的 SQL 语言,则会产生一个异常:句法错误或违反访问规则。

在访问规则中,术语应该定义成功应用一般规则所需满足的条件。当应用一般规则时,如果有任何

这样的条件未被满足,则会产生一个异常:句法错误或违反访问规则。

在符合性规则中,术语应该定义当命名特征不要求被支持时需要为真的条件。

6.3.3.3 规则求值顺序

符合性实现并不要求严格按照一般规则中定义的动作序列来执行,只要执行对于 SQL 数据与模式、宿主参数与宿主变量以及 SQL 参数与 SQL 变量产生的效果与该序列的效果相同。术语有效地用来强调实现可能通过其他方法获得其效果的动作。

包含句法元素的句法规则和访问规则同时也作为句法规则和访问规则被有效地应用于包含它的句法元素。包含句法元素的一般规则,可在一般规则之前被有效地应用于包含它的句法元素。

在操作符的优先级是通过 GB/T 12991 的格式或括号确定的场合,那些操作符按该优先级规定的顺序被有效地应用。

在优先级不是通过格式或括号确定的场合,表达式的有效求值通常从左到右执行。然而,表达式实际上是否从左到右求值依赖于实现,尤其当操作数或操作符可能导致状态的产生,或者表达式的结果不需对表达式的所有部分完全求值就能确定的情况下更是如此。

通常,如果某个句法元素包含一个以上的其他句法元素,则该句法元素的产生式中早出现的被包含元素的一般规则先于晚出现的被包含元素的一般规则被应用。

例如,在下面的产生式中:

$$\langle A \rangle ::= \langle B \rangle \langle C \rangle$$

$\langle A \rangle$ 、 $\langle B \rangle$ 和 $\langle C \rangle$ 的句法规则和访问规则可同时有效地被应用。 $\langle B \rangle$ 的一般规则在 $\langle C \rangle$ 的一般规则之前被应用, $\langle A \rangle$ 的一般规则在 $\langle B \rangle$ 和 $\langle C \rangle$ 的一般规则之后被应用。

如果表达式或搜索条件的结果不依赖于该表达式或搜索条件的某个部分的结果,则该表达式或搜索条件的那个部分被称为是非本质的。如果 SQL 调用函数的调用结果是确定的且不可能修改 SQL 数据,则它是非本质的;否则,它是本质的还是非本质的是实现定义的。

如果与非本质部分相关的访问规则不被满足,则无论非本质部分实际上是否被求值,均产生异常状态句法错误或违反访问规则。如果非本质部分的求值可能会引起产生一个异常状态,则是否产生异常状态依赖于实现。

在计算表达式的结果期间,SQL 实现可能产生用于确定结果的一个或多个中间结果。包含中间结果之场所的声明类型是实现相关的。

6.3.3.4 条件规则

条件规则是用“IF”和“CASE”约定指明的。用“CASE”约定指明的规则包括一个使用“IF”约定的条件子规则的清单。第一个条件为真的“IF”子规则就是该“CASE”规则的有效子规则。“CASE”规则的最后一个子规则可指明为“OTHERWISE”,如果“CASE”规则中前面的“IF”子规则中没有一个条件为真,则这个子规则就是“CASE”规则的有效子规则。

6.3.3.5 句法替换

在句法规则和一般规则中,短语“X 是隐含的”表示句法规则和一般规则是在元素 X 实际上已经被说明的情况下解释的。在某一给定条款的句法规则内,可以知道该元素是显式规定的还是隐含的。

在句法规则和一般规则中,短语“下列 $\langle X \rangle$ 是隐含的:Y”表示该句法规则和一般规则是在包含 Y 的句法元素 $\langle X \rangle$ 实际上已被说明的情况下解释的。

在句法规则和一般规则中,短语“前者等价于后者”表示该句法规则和一般规则是在元素中前者的所有实例均为后者的实例的情况下解释的。

如果在一个条款中引用了 BNF 非终结符,而没有指明它在该条款所定义的 BNF 产生式中是如何被包含的,则有以下情况:

——如果 BNF 非终结符是在该条款中自定义的,则该引用应该被假定是在定义它的产生式的左边出现的那个 BNF 非终结符;

——否则,该引用应该被假定是立即包含着该 BNF 非终结符的 BNF 产生式。

6.3.3.6 其他术语

某些句法规则定义了一些术语,如 T1,表示命名表或非命名表。这些术语被用为表名或相关名。在被用为相关名时,它既不表示对指定的表实际上定义了任何新的相关名,也不会影响任何实际相关名的作用域。

如果 SQL 语句 S2 是一条〈调用语句〉CS,且 SQL 语句 S1 是〈SQL 调用例程〉包含的最外层 SQL 语句,而〈SQL 调用例程〉是 CS 中所含〈例程调用〉的目标例程,则称 S1 是作为执行 S2 的直接结果而被执行的。

如果 SQL 语句 S1 是在由执行 SQL 语句 S2 而激活的一个触发器执行上下文中被执行的,则称 S1 是作为执行 S2 的间接结果而被执行的。

值 P 是值 W 的一部分,当且仅当:

- W 是一个表,而 P 是 W 中的一行;
- W 是某一行,而 P 是 W 中的一个字段;
- W 是一个集合,而 P 是 W 中的一个元素;
- P 是某个值的一部分,而该值是 W 的一部分。

如果一个值包含几个部分,则该值的一个实例也包含几个部分;因此,它所在的场所也包含几个部分,而每个部分也是一个场所。

项 X 是项 Y 的一部分,当且仅当:

- Y 是一行,而 X 是 Y 的某个列;
- Y 是一个例程调用,而 X 是 Y 的一个 SQL 参数;
- Y 是一个用户定义类型的实例,而 X 是 Y 的一个属性;
- 存在一个项 X2,且 X 是 X2 的一部分,而 X2 是 Y 的一部分。

GB/T 12991 中其他部分定义的附加术语,仅在该部分中使用。

6.3.3.7 异常

除非另有所指,一般规则或其他地方的短语“产生异常状态:”(后面跟随一个状态名)表示:

- 语句的执行是不成功的;
- 在 6.3.3.8“异常状态下不终止的一般规则”中没有说明的一般规则的应用可能被终止;
- 诊断信息变得有效;
- 语句的执行对 SQL 数据或模式无影响。

除非 GB/T 12991 有明确定义,否则以异常状态终止的 SQL 语句对任何赋值目标和 SQL 描述符区的影响是实现相关的。

一般规则或其他地方的短语“产生完成状态:”(后随一个状态名)表示,一般规则的应用未被终止,且诊断信息变得有效;除非同时产生了一个异常状态,否则语句的执行是成功的。

如果因一个语句执行出现多个状态,则有关多个状态的诊断信息是否有效是实现相关的。有关状态参数值的优先规则,见 GB/T 12991.2 中 4.29.2。

6.3.3.8 异常状态下不终止的一般规则

在异常状态产生的地方,下列条款的一般规则并不终止:

- GB/T 12991.2 中 10.4“〈例程调用〉”;
- GB/T 12991.2 中 13.5“〈SQL 过程语句〉”;
- GB/T 12991.2 中 14.27“触发器的执行”;
- GB/T 12991.2 中 21.1“〈直接 SQL 语句〉”;
- GB/T 12991.4 中 13.1“〈复合语句〉”;
- GB/T 12991.4 中 13.2“〈句柄声明〉”。

6.3.4 描述符

描述符是一个概念性结构的数据集合,它定义指定类型的对象。描述符的概念用来说明 SQL 的语义。在任何 SQL 环境中并不一定存在任何特定形式的任何描述符。

某些 SQL 对象离开了其他 SQL 对象的上下文是不能存在的。例如,离开了表的上下文,列是不能存在的。每个这样的对象都是由它自己的描述符来独立描述的,并且称属主对象(例如:表)的描述符包括每个从属对象(例如:列或者表约束)的描述符。反之,称该从属对象的描述符被包括在属主对象的描述符中。如果某个对象 A 的描述符包括某个对象 B 的描述符,且 B 的描述符一般包括某个对象 C 的描述符,则对象 A 的描述符一般包括 C 的描述符。

在另一些情况下,除非其他 SQL 对象存在,否则某些 SQL 对象是不可能存在的,并且它们之间没有包括关系。例如,如果某个断言所引用的表不存在,则 SQL 不允许该断言存在。因此,断言描述符依赖于或取决于一个或多个表描述符(等价地,断言依赖于或取决于一个或多个表)。一般而言,描述符 D1 可被称作取决于或依赖于某个描述符 D2。

如果对象 A 的描述符依赖于对象 B 的描述符,且 B 的描述符一般依赖于对象 C 的描述符,则 A 的描述符一般依赖于 C 的描述符。

SQL 语句的执行可能导致很多描述符的创建。作为 SQL 语句的结果而创建的 SQL 对象可能依赖于其他描述符,这样的描述符是作为该 SQL 语句的执行结果而创建的。

注 6: 这是〈模式定义〉SQL 语句的特殊情况。例如,一个〈模式定义〉可能包含很多〈表定义〉,而〈表定义〉中又包含很多〈表约束〉。在一个〈表定义〉中的〈表约束〉可以引用由另外的〈表定义〉所创建的第二个表,而第二个表又包含对第一个表的引用。在〈模式定义〉执行期间,只有创建了所有必需的描述符,描述符之间的相互依赖才是有效的。

表示一个 SQL 对象依赖于另一个 SQL 对象的方式有两种。大多数情况下,依赖 SQL 对象的描述符被称为“包含它所依赖的 SQL 对象名”。这时,“名”可理解为“标识描述符的足够信息”。另一种情况下,依赖 SQL 对象的描述符可被称为包含它所依赖的 SQL 对象的文本(如〈查询表达式〉,〈搜索条件〉)。不过在这种情况下,究竟包含的是实际文本(其中缺省和隐式部分可能被变为显式)还是其对应的语法分析树则无关紧要,描述符的有效性明确地“依赖于”其引用对象的描述符的存在。

列“基于”某个域,等价于说列“依赖于”这个域。

当企图撤销一个 SQL 对象时,如果存在其他描述符依赖于该对象的描述符,则此撤销操作可能失败,具体结果取决于撤销行为是如何规定的。如果要撤销的描述符被包含在其他描述符中,撤销操作也可能会失败。撤销描述符将导致自身包括的所有描述符的撤销,但对于它所依赖的描述符没有影响。

由描述符所描述的某些 SQL 对象的实现需要一些本标准中没有规定的对象存在。在需要这种对象的地方,当相关的描述符被创建时,对象就会被创建;当相关的描述符被撤销时,对象就会被撤销。

6.3.5 GB/T 12991 中各部分的关系

除 GB/T 12991 的本部分和 GB/T 12991.2 外的其他各部分都依赖于 GB/T 12991.1、GB/T 12991.2,并作为增加部分被引用。每个增加部分在使用时就像它与 GB/T 12991 的内容已融合在一起。本条描述用于融合的相关约定。

有关条款的修改调整是通过“在 TC 中”的短语来指明的。

6.3.5.1 新增的和修改的章、条、表及附录

如果 GB/T 12991 任一增加部分的某章(除第 1 章“范围”和第 2 章“规范性引用文件”外)、条、表或附录的名字与 GB/T 12991.1、GB/T 12991.2、GB/T 12991.3、GB/T 12991.4 或 GB/T 12991.11(除非该增加部分本身是 GB/T 12991.3、GB/T 12991.4 或 GB/T 12991.11)中的某章、条、表或附录的名字相同,则它分别补充 GB/T 12991.1、GB/T 12991.2、GB/T 12991.3、GB/T 12991.4 或 GB/T 12991.11 中的该章、条、表或附录,而不管该章、条、表或附录的编号、字母或位置是不是相对应。典型的做法是通过增加或替换段落、格式项、表项或规则来实现。

被修改的表中的行通常为要插入到对应表中的新行,不过在极少数情况下,是将原表中已有的一行替换为增加部分中的表中的一行。当对应表的第一列的值相同的时候,就要求进行这种替换。

在每个增加部分中,每章、条、表及附录与 GB/T 12991.1 和/或 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.4 和/或 GB/T 12991.11 中对应的章、条、表或附录之间的关系按如下方法表示:

——在增加部分中:形如“本对象修改 GB/T 12991. n 中的对象 nn. nn“xxxxx””的陈述直接跟随该对象标题。

——在上一步的陈述所引用的部分中:形如“本对象由 GB/T 12991. m 中的对象 mm. mmm“yyyyy”修改”的陈述直接跟随该对象标题。

——该对象可以为章、条、表或附录。

如果增加部分的某章、条、表或附录的名字与 GB/T 12991.1 和/或 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.4 和/或 GB/T 12991.11 中的章、条、表或附录的名字不同,则它提供的语言说明是特别针对那个部分的。属于某个已经标识为新的章条的条款或表自然是新的,没有再作标记。

每个增加部分的章、条或附录按照它们在融合文档中被确定的顺序出现。当缺少关于其位置的显式说明时,任何新章、条或附录按下述方法定位:在 GB/T 12991.1 和/或 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.4 和/或 GB/T 12991.11 中定位前一条其名字与 GB/T 12991 增加部分的对应章、条或附录的名字相同的章、条或附录,新的章、条或附录直接跟随在该章、条或附录的后面。如果存在多个新章、条或附录,且它们之间没有修改已有章、条或附录的中间章、条或附录,则这些新章、条或附录跟随在前一个其名字匹配的章、条或附录后依次出现。

增加部分对 GB/T 12991.3 和/或 GB/T 12991.4 和/或 GB/T 12991.11 中的章、条、表或附录进行修改时,按下列顺序实施修改:

- 1) 实施该增加部分对 GB/T 12991.3 的全部修改;
- 2) 实施该增加部分对 GB/T 12991.4 的全部修改;
- 3) 实施该增加部分对 GB/T 12991.11 的全部修改;
- 4) 实施 GB/T 12991.11 对 GB/T 12991.2 的全部修改(包括第 3 步中所有增加、扩展或替换的结果);
- 5) 实施 GB/T 12991.3 对 GB/T 12991.2 的全部修改(包括第 1 步中所有增加、扩展或替换的结果);
- 6) 实施 GB/T 12991.4 对 GB/T 12991.2 的全部修改(包括第 2 步中所有增加、扩展或替换的结果);
- 7) 实施该增加部分对 GB/T 12991.2 的全部修改,注意最后一步的修改可能扩展或替换第 4 步、第 5 步、第 6 步的修改。

多个增加部分对 GB/T 12991.2、GB/T 12991.3、GB/T 12991.4 和/或 GB/T 12991.11 实施的修改不会相互影响。由某一个增加部分所作的修改只对该部分的语言说明有影响,而这些说明不会受到任何其他增加部分所作修改的影响。

6.3.5.2 功能

在修改的条款中,用来自修改条款中的功能完全替换原条款中的功能。

6.3.5.3 新的和修改的格式项

在修改的章条中,定义一个 BNF 非终结符(即:该 BNF 非终结符出现在记法::=的左边)的格式项,或是修改其定义出现在 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中的某个格式项;或是替换其定义出现在 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中的某个格式项;或是定义一个 GB/T 12991.2 和/或

GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中没有的某个新格式项。

在增加部分中,对于修改其定义出现在 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中的格式项的那些格式项,是通过“格式注释”来标识的,例如:

〈修改项〉 ::= !! 来自 GB/T 12991.2 的全部替换项
| 〈新替换项〉

相比而言,完全替换 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中的格式项的那些格式项,具有与 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中的 BNF 非终结符相同的 BNF 非终结符,但并未声明其包含 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中的任何替换项。

在 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中没有任何对应项的那些新格式项在增加部分是不作区别的。

新条框中的格式项不作标记。

6.3.5.4 新的和修改的段落和规则

在修改的条款中,每个段落或规则都作了标记,以表明其是对 GB/T 12991.1 和/或 GB/T 12991.2 和/或 GB/T 12991.3 和/或 GB/T 12991.10 和/或 GB/T 12991.11 中段落或规则的修改,还是由该增加部分新增的段落或规则。

段落或规则的修改是通过封装在条形框中的指示短语来标识的。

代替第 5 段 表示随后的正文用于替换条标题后的语句所标识的部分中对应条的第 5 段。

代替 SR6)b)ii) 表示随后的正文用于替换条标题后的语句所标识的部分中对应条的句法规则 6)b)ii)。

扩展 SR3) 表示随后的正文用于扩展或增强句法规则 3)。在大多数实例中,该扩展是为了支持新句法而新加的替换项。

没有限定的“替换”意味着被指示层次的整个段或规则都要被替换。例如,代替 SR6)b) 意味着句法规则 6)b)和由它包含的任何子规则都要被替换。如果包括引导语这个术语,则替换的范围为该段或规则中从开始直到(但不包括)第一个列项的正文部分。例如,代替 SR1)b)的引导语 意味着下列规则中只有斜体正文要被替换。

- 1) 句法规则 1)的引导语
- 2) a) 句法规则 1)a)的正文
 - b) 句法规则 1)b)的引导语
 情况如下:
 - i) 语法规则 1)b)i)的正文
 - ii) 语法规则 1)b)ii)的正文
- c) 句法规则 1)c)的正文

增加部分的新段或规则作了标记,以指示插入的地方。

插在第 2 段之前 表示随后的正文应直接插到条标题后的语句所标识的部分中对应条的第 2 段之前。

插在 GR4)之前 表示随后的正文应直接插到条标题后的语句所标识的部分中对应条的一般规则 4)之前。

如果没有指定具体插入点,如插入此段或插入此 GR,则随后的正文应添加到条标题后的语句所标识的部分中对应条中相应部分的末尾。

每个条中的段落从 1 开始编号,列项的内容作为同一段的正文部分来看待。编号注释不作为段统计。

在这些指示短语中,“SR”表示“句法规则”,“AR”表示“访问规则”,“GR”表示“一般规则”,“CR”表示“符合性规则”,“Desc”表示“描述”,“Func”表示“功能”。

新章条中的所有的段、格式项和规则也是新的,因此不作标记。

6.3.5.5 修改的附录

在修改的附录中,修改附录段的引导语被废止。

注 7:这将导致只有表和列项保持不变。

所有其他的正文和/或表被融入原附录的正文和/或表中,并考虑原附录的原始段所隐含的分组和理序。

注 8:正文中引用的章条将被调整为归并后的正文中的引用章条的值。

6.3.5a 作为子例程使用的条款

在本标准的各部分中,某些条款没有定义调用其语义的显式语法。这种条款(称为子例程条款)的典型情况是列出了其他一个或多个条款所需要的,且被确定为由那些条款中的规则所调用的规则。在少数情况下,这些没有显式语法的条款被确定为由其他标准和/或通过使用实现定义的机制来调用。

换句话说,这些条款中的规则就像是一种由其他条款、标准或实现定义的机制调用的说明性的子例程。这些子例程条款典型的说明方式是要求其调用者向其传递信息。这些必须传递的信息被表示为该子例程条款的参数,并且该信息必须由该子例程条款的调用者以变元的形式传递。

对于子例程条款的每次调用都必须为被调用的子例程条款的每个参数显式提供信息。如果调用一个子例程条款时没有为每个要求的参数提供信息,其结果是实现相关的。

6.3.6 索引排版格式

有关 GB/T 12991 的索引,遵循下述约定:

- 黑体索引项指出的是该词、短语、BNF 非终结符定义所在的页;
- 斜体索引项指出的是该 BNF 非终结符用作格式项所在的页;
- 既非黑体也非斜体的索引项指出的不是该词、短语、BNF 非终结符定义所在的页,而是其用于除格式外的其他部分(例如,标题、功能、句法规则、访问规则、一般规则、符合性规则、表或其他描述性文本)所在的页。

6.3.7 特征 ID 和特征名

特征包括标准定义的特征和实现定义的特征。

标准定义的特征在 GB/T 12991 的各部分中定义,实现定义的特征由 SQL 实现定义(见 8.4.8)。

特征通过特征 ID 和特征名来引用。特征 ID 要么由一个字母和三个数字组成,要么由一个字母、三个数字、一个连字符和一个或两个数字组成。包含连字符和附加数字的特征 ID 表示“子特征”,子特征有助于定义完整的特征,该特征以不带连字符的特征 ID 来表示。

字母为“V”的特征 ID 是为实现定义的特征所保留的。

一个特征子特征集列出了该特征中值得注意的特殊情况,但并不要求完备。另外,一个特征各子特征之间不必是互斥的,在某些情况下,一个子特征可能包含另一个子特征。

只有整个特征才能用于说明符合性要求。

标准定义的特征要么是某个部分的必选特征,要么是由符合性规则定义的可选特征。标准定义的特征的特征 ID 是稳定的,且保持不变。

为方便起见,GB/T 12991 中某个部分定义的所有特征被收集到该部分的附录中。例如,GB/T 12991.2 中定义的特征能够在 GB/T 12991.2 的附录 F“SQL 特征分类”中找到。

通常情况下,符合性规则放在定义受特征控制的 BNF 非终结符的条款中。在非终结符的使用被控制于特定环境的情况下,符合性规则放在使用非终结符的地方。如果非终结符的使用受特征控制的事

实能够根据其他条款的句法规则和符合性规则的检查而推断出,符合性规则也放在使用非终结符的地方,且通常是冗余的。符合性规则的其他冗余则能避免就避免。

6.4 数据库语言 SQL 的对象标识符

数据库语言 SQL 具有一个采用 GB/T 16262.1 中机制定义的对象标识符,它标识了一个 SQL 实现的特性。

功能

数据库语言 SQL 的对象标识符标识了 SQL 实现的特征,以区别于开放系统环境中的其他实体。

注 9: SQL 用户可从信息模式中得到等价的信息。

格式

〈SQL 对象标识符〉 ::= 〈SQL 源〉 〈SQL 变种〉

〈SQL 源〉 ::= 〈参数 1〉 〈参数 2〉 〈参数 3〉

〈参数 1〉 ::= iso | 1 | iso 〈左圆括号〉 1 〈右圆括号〉

〈参数 2〉 ::= standard | 0 | standard 〈左圆括号〉 0 〈右圆括号〉

〈参数 3〉 ::= 9075

〈SQL 变种〉 ::= 〈SQL 版本〉 〈SQL 符合性〉

〈SQL 版本〉 ::= 〈版本号〉 | 〈版本日期〉 〈左圆括号〉 〈版本号〉 〈右圆括号〉

〈版本号〉 ::= 4

〈版本日期〉 ::= edition2003

〈SQL 符合性〉 ::= 〈部分〉 [〈包〉] [〈特征集〉]

〈部分〉 ::= 〈部分 n〉

〈部分 n〉 ::= 〈部分 n 否〉 | 〈部分 n 是〉

〈部分 n 否〉 ::= 0 | Part-n-No 〈左圆括号〉 0 〈右圆括号〉

〈部分 n 是〉 ::= 1 | Part-n-Yes 〈左圆括号〉 1 〈右圆括号〉

〈包〉 ::= 〈包 PKGi〉 …

〈包 PKGi〉 ::= 〈包 PKGi 是〉 | 〈包 PKGi 否〉

〈包 PKGi 是〉 ::= 1 | Package-PKGi-Yes 〈左圆括号〉 1 〈右圆括号〉

〈包 PKGi 否〉 ::= 0 | Package-PKGi-No 〈左圆括号〉 0 〈右圆括号〉

〈特征集〉 ::= 〈特征 FEATi〉…

〈特征 FEATi〉 ::= 〈特征 FEATi 是〉 | 〈特征 FEATi 否〉

〈特征 FEATi 是〉 ::= 1 | Feature-FEATi-Yes 〈左圆括号〉 1 〈右圆括号〉

〈特征 FEATi 否〉 ::= 0 | Feature-FEATi-No 〈左圆括号〉 0 〈右圆括号〉

注 10: 标识 GB/T 12991 标准不同版本的对象标识符的变种是通过〈SQL 版本〉的不同值来区分的。对于标准的不同版本,〈SQL 符合性〉的结构和解释可能是不同的。〈SQL 版本〉的区分值和〈SQL 符合性〉的相关结构和解释,在其应用的 GB/T 12991 的各版本中定义。

句法规则

- 1) 〈部分 n 是〉意味着 GB/T 12991. n 的符合性被声明;
- 2) 〈部分 n 否〉意味着 GB/T 12991. n 的符合性未被声明;
- 3) 〈包 PKGi 是〉意味着由 PKGi 标识的可选包的符合性被声明,其中 PKGi 是 GB/T 12991 的本部分或 GB/T 12991 的某个轮廓中描述的一个可选包的包 ID;
- 4) 〈包 PKGi 否〉意味着由 PKGi 标识的可选包的符合性未被声明;
- 5) 〈特征 FEATi 是〉意味着由 FEATi 标识的可选特征的符合性被声明,其中 FEATi 是 GB/T 12991 的某部分中描述的一个可选特征的特征 ID;

注 11: 有关特征的定义,见 6.3.7。

6) 〈特征 FEAT_i 否〉意味着由 FEAT_i 标识的可选特征的符合性未被声明。

7 GB/T 12991 中各部分的附录

GB/T 12991 中每个部分的附录都是资料性的。每个附录的内容提供附加的信息,某些内容重申了标准正文中某个地方所声明的内容。

7.1 实现定义的元素

GB/T 12991 中的每部分都包含这样一个附录,该附录列出了该部分说明的允许 SQL 实现之间不同但要求特定 SQL 实现的实现者说明的每个 SQL 元素及其处理。

7.2 实现相关的元素

GB/T 12991 中的每部分都包含这样一个附录,该附录列出了该部分提到而未作规定、因而允许 SQL 实现之间不同、但不要求特定 SQL 实现的实现者说明的每个 SQL 元素及其处理。

7.3 不推荐的特征

GB/T 12991.2 和每个增加部分都包含这样一个附录,该附录列出了该部分说明的、但是在未来版本中可能不再规定的每个 SQL 元素及其处理。

7.4 与以前版本的不兼容性

GB/T 12991.2 和每个增加部分都包含这样一个附录,该附录列出了该部分的以前版本说明过、而当前版本不是按相同方法说明的每个 SQL 元素及其处理。导致这种不兼容性的常见原因是在语言中增加保留字,这将使得它们在符合 GB/T 12991 早期版本的 SQL 语言中的使用变得无效。

8 符合性

被声明的符合性可以有多种类型。每个符合性声明均应包括最小符合性声明。另外,还可以声明零个或多个增加部分、零个或多个可选特征及零个或多个包的符合性。

8.1 最小符合性

每个符合性声明均应包括最小符合性声明,最小符合性声明定义为满足 GB/T 12991.2 和 GB/T 12991.11 中说明的符合性要求。最小符合性声明应包括 GB/T 12991.2 和 GB/T 12991.11 的符合性要求所要求的语句。

仅声明最小符合性声明的 SQL 语言称为核心 SQL。

8.2 部分的符合性

GB/T 12991 某个部分的符合性声明隐含着支持那个部分定义的所有必选特征。

此外,每个对 GB/T 12991 某个部分的符合性声明还应满足该部分规定的符合性要求。

对 GB/T 12991 某个增加部分的符合性声明,应包括声明了符合性的 GB/T 12991 中那个部分的符合性要求所要求的声明。

8.3 特征的符合性

除了 GB/T 12991 中每个部分(包括 GB/T 12991.2 和 GB/T 12991.11)的符合性所强制要求的那些特征外,GB/T 12991 的任何部分都可定义一些可选特征。这些特征由特征 ID 来标识,且受符合性规则控制(见 6.3.7)。

一个可选特征 FEAT 由通过放宽所选择的一些符合性规则来定义。每个符合性规则的开始,都由短语“如果没有特征 FEAT‘特征名’,…”提示。一个应用指出该应用所要求的 SQL 特征集,该应用的 SQL 语言应注意到所有符合性规则的限制,除了明显放宽的这些特征外。反过来,符合性 SQL 实现应标识 SQL 实现所支持的那些 SQL 特征。一个 SQL 实现应能处理任何这样的应用,该应用要求的特征是 SQL 实现所支持特征的子集。

一个特征 FEAT₁ 可能隐含另一个特征 FEAT₂。声明支持 FEAT₁ 的 SQL 实现也应支持由 FEAT₁ 所隐含的每个特征 FEAT₂。反过来,一个应用只需要指出它要求 FEAT₁,则假定包括了由

FEAT1 所隐含的每个 FEAT2。由某些特征所隐含的特征清单在名为“……的隐含特征关系”的表中列出,该表位于 GB/T 12991 的每个部分中名为“符合性”的章中。要注意的是,有些特征隐含着多个其他特征,有些特征则隐含着其他部分所定义的特征。GB/T 12991 的每个可选特征的符合性声明应满足符合性被声明的所有部分的符合性要求,就像控制该特征的符合性规则不存在一样。

句法规则和一般规则可以根据一条 SQL 句法来定义另一条。这种变换用于定义变换句法的语义,并且在检查可用符合性规则后就被有效地实施。变换可以使用一个 SQL 特征的 SQL 句法来定义另一个 SQL 特征。这些变换用来定义句法的行为,但与被如此定义的特征允许或禁止的特征句法无任何牵连。一个符合性 SQL 实现只需处理其声明支持的特征集的符合性规则所定义的未转换句法,虽然带有转换所隐含的语义。

8.4 SQL 包的符合性

包是 GB/T 12991 的一个可选特征组。它与核心 SQL 一起为某个应用领域或实现环境提供平台。包由包 ID 来标识。

对 GB/T 12991 的某个包的每个符合性声明应满足所有部分和组成该包的所有可选特征的符合性要求。

本条规定了几个这样的包,预计这些包的一些特征可能在 GB/T 12991 范围之外规定。

表 3“SQL 包”包含由 GB/T 12991 定义的 SQL 语言包清单。

表 3“SQL 包”中的列“包 ID”说明表中所包含的包的正规标识。包 ID 是稳定的且保持不变。

表 3“SQL 包”中的列“包描述”包含与包 ID 值相关联的包的简要描述。

表 3 SQL 包

包 ID	包 描 述
PKG001	增强日期时间设施
PKG002	增强完整性管理
PKG004	PSM
PKG006	基本对象支持
PKG007	增强对象支持
PKG008	主动数据库
PKG010	OLAP

8.4.1 增强日期时间设施

包“增强日期时间设施”由下列 SQL 语言特征组成,这些特征在 GB/T 12991 各部分的 SQL 特征分类附录中说明:

- 特征 F052“时间间隔和日期时间算法”;
- 特征 F411“时区说明”;
- 特征 F555“增强秒精度”。

8.4.2 增强完整性管理

包“增强完整性管理”由下列 SQL 语言特征组成,这些特征在 GB/T 12991 各部分的 SQL 特征分类附录中说明:

- 特征 F191“引用删除动作”;
- 特征 F521“断言”;
- 特征 F701“引用更新动作”;
- 特征 F491“约束管理”;
- 特征 F671“CHECK 约束中的子查询”;

- 特征 T201“引用约束的可比较数据类型”；
- 特征 T211“基本触发器能力”；
- 特征 T212“增强触发器能力”；
- 特征 T191“引用动作 RESTRICT”。

8.4.3 PSM

包“PSM”由下列 SQL 语言特征组成,这些特征在 GB/T 12991 各部分的 SQL 特征分类附录中说明:

- 特征 T322“SQL 调用函数和 SQL 调用过程的重载”；
- 特征 P001“存储模块”；
- 特征 P002“计算完备性”；
- 特征 P003“信息模式视图”。

8.4.4 基本对象支持

包“基本对象支持”由下列 SQL 语言特征组成,这些特征在 GB/T 12991 各部分的 SQL 特征分类附录中说明:

- 特征 S032“基本结构类型”；
- 特征 S041“基本引用类型”；
- 特征 S051“创建类型表”；
- 特征 S151“类型谓词”；
- 特征 T041“基本 LOB 数据类型支持”。

8.4.5 增强对象支持

包“增强对象支持”由包“基本对象支持”的所有特征和下列 SQL 语言特征组成,这些特征在 GB/T 12991 各部分的 SQL 特征分类附录中说明。

- 特征 S024“增强结构类型”；
- 特征 S043“增强引用类型”；
- 特征 S071“函数和类型名解析中的 SQL 路径”；
- 特征 S081“子表”；
- 特征 S111“查询表达式中的 ONLY”；
- 特征 S161“子类型处理”；
- 特征 S211“用户定义转换函数”；
- 特征 S231“结构类型定位器”；
- 特征 S241“转换函数”。

8.4.6 主动数据库

包“主动数据库”由下列 SQL 语言特征组成,这些特征在 GB/T 12991 各部分的 SQL 特征分类附录中说明:

- 特征 T211“基本触发器能力”。

8.4.7 OLAP

包“OLAP”由下列 SQL 语言特征组成,这些特征在 GB/T 12991 各部分的 SQL 特征分类附录中说明:

- 特征 T431“增强分组能力”；
- 特征 T611“基本 OLAP 操作符”。

8.4.8 扩展和选项

除了 GB/T 12991 规定的那些特征外,SQL 实现可以提供实现定义的特征,并将其加到保留字清单中。

注 12: 如果增加了保留字,可能无法正确地处理符合性 SQL 语句。

SQL 标记符是否标记实现定义的特征是实现定义的。

注 13: SQL 标记符可以使用本标准没有定义的任何特征 ID 标记实现定义的特征,但不保证本标准将来的某个版本不使用该特征 ID 作为标准定义的特征。

注 14: 由 SQL 标记符所标记的实现定义的特征可以标识来自多个 SQL 实现的实现定义的特征。

注 15: 不同 SQL 标记符为实现定义的特征分配的特征 ID 可能不同。

注 16: SQL 实现可选择(但不要求)使实现定义的特征在信息模式的 SQL_FEATURES 视图中可见。

SQL 实现可以提供对增加的实现定义的 SQL 调用例程或实现定义的 SQL 调用例程变元值的支持。

SQL 实现可以提供一些用户选项来处理非符合性 SQL 语句或例程调用。SQL 实现可以提供一些用户选项来处理 SQL 语句或例程调用,以便产生不同于 GB/T 12991 中各个部分所规定的结果。

只有当用户选项显式要求时,才产生这样的结果。增加的扩充和选项可以在 GB/T 12991 的增加部分中说明。

8.5 SQL 标记符

SQL 标记符是实现提供的设施,它能够标识符合性 SQL 实现可能提供的 SQL 语言扩展或其他 SQL 处理替换项(见 8.4.8“扩展和选项”)。

SQL 标记符旨在帮助 SQL 程序员在本标准的不同级别下,编写便于在不同符合性 SQL 实现之间移植和互操作的 SQL 语言。

SQL 标记符旨在影响 SQL 语言的静态检查,不要求探测在运用一般规则之后才能确定的那些扩展。

就实现而言,SQL 实现只需标记没有错误的 SQL 语言。

注 17: 如果一个系统正在处理包含错误的 SQL 语言,那么,在单条语句内要确定什么是错误什么是扩展是很困难的。作为一种可能性,一个实现可以选择按两步来检查 SQL 语言:首先通过正常的句法分析器;其次通过 SQL 标记符。第一个步骤对实现不能处理或不能识别的非标准 SQL 语言产生错误消息;第二个步骤只处理就实现而言没有任何错误的 SQL 语言,它同时探测和标记能够由实现处理的所有非标准 SQL 语言。任何这样的两步处理对最终用户来说都应该是透明的。

SQL 标记符允许应用程序员来标识符合性 SQL 实现提供的符合性 SQL 语言,该符合性 SQL 语言在不同处理环境下的执行可能是不同的。如果 SQL 语言从非符合性 SQL 处理环境迁移到符合性 SQL 处理环境,还必须提供一个工具用以标识不得不修改的 SQL 元素。

SQL 标记符提供一个或多个下述“标记级”选项:

- 核心 SQL 标记;
- 部分 SQL 标记;
- 包 SQL 标记。

提供这些选项之一的 SQL 标记符应能够标识违反 SQL 语言的指定子集的 SQL 语言结构。“核心 SQL 标记”使用的 SQL 语言子集为核心 SQL;“部分 SQL 标记”使用的 SQL 语言子集为核心 SQL 加上符合 GB/T 12991 的各指定部分所要求的那些特征。“包标记”使用的 SQL 语言子集为核心 SQL 加上符合 GB/T 12991 的各指定包所要求的那些特征。

SQL 标记符还提供“SQL 特征标记”。“SQL 特征标记”指出,除了指定的 SQL 语言子集外,还需要哪些可选特征来保证 SQL 语言的符合性。

SQL 标记符提供一个或多个下述“范围检查”选项:

- 仅限句法;
- 目录检查。

在仅检查句法的情况下,SQL 标记符只分析碰到的 SQL 语言,它检查无须访问信息模式便能确定的任何违反句法规则的情况,不必探测依赖于句法元素的数据类型的非法情况,即使理论上这样的非法

情况根据句法就能推断出来。

在目录检查的情况下,SQL 标记符假设了定义模式信息的可用性,并检查违反所有句法规则的情况。例如,某些句法规则在数据类型上施加了一些限制,该标记符选项会标识放宽这些限制的扩展。为安全起见,本选项应通过特定的信息模式检查定义模式。标记符不必执行或模拟任何〈模式定义语句〉或〈模式操纵语句〉的执行。

8.6 符合性声明

符合 GB/T 12991 的声明应包括下列所有声明:

- 1) 最小符合性声明;
- 2) 零个或多个增加部分的符合性声明;
- 3) 零个或多个包的符合性声明;
- 4) 零个或多个可选特征的符合性声明。

注 18: 除本章的要求外,GB/T 12991 的每个部分均说明了该部分的符合性声明需要陈述的内容。

8.6.1 对 SQL 应用的要求

术语“SQL 应用”这里用来表示一组编译单元的汇集,每个编译单元均采用某种标准程序设计语言,其中包含一个或多个:

- SQL 语句;
- SQL/CLI 例程的调用;
- 外部调用过程的调用。

符合性 SQL 应用不应包含句法错误,并满足下列要求:

- 每个 SQL 语句或 SQL/CLI 调用在句法上都符合 GB/T 12991;
- 模式内容满足 SQL 应用的要求;
- SQL 数据与模式内容相符;
- 不允许用户提交有句法错误的直接执行 SQL 语句。

符合性 SQL 应用不应使用任何附加特征,或超出所声明的符合性级别的特征。

SQL 应用的符合性声明还应说明:

- 正确运行依赖于哪些实现定义的元素和动作;
- 需要由用户提供什么样的模式内容。

8.6.2 对 SQL 实现的要求

符合性 SQL 实现应提供一个对象标识符(见 6.4),该对象标识符陈述 GB/T 12991 中被声明了符合性的那些部分、包和特征。

符合性 SQL 实现应根据相关的一般规则、定义和描述处理符合性 SQL 语言。

符合性 SQL 实现应根据相关的定义和一般规则处理符合性例程调用。

SQL 应用的符合性声明还应说明:

- GB/T 12991 规定为实现定义的各个元素和动作的定义。

提供附加设施或超出“核心”规定范围的设施的符合性 SQL 实现应提供 SQL 标记符。

附录 A

(资料性附录)

SQL 国际标准的维护和解释

ISO/IEC JTC1 提供对 JTC1 标准所进行的修改、维护和解释的正式规程。JTC1 规程“国际标准的维护”中的第 14 章,规定了建立和处理“缺陷报告”的过程。缺陷报告可能导致现行国际标准的技术性勘误、修正、解释或其他注释的产生。

自 ISO/IEC 9075:1999 出版以来,已出版了两个技术勘误。

- 1) ISO/IEC 9075:1999/Cor. 1——该勘误表包含对 GB/T 12991.1:1999、GB/T 12991.2:1999、ISO/IEC 9075-3:1999、ISO/IEC 9075-4:1999、ISO/IEC 9075-5:1999 的更正累积。
- 2) ISO/IEC 9075(部分 1 到部分 5):1999/Cor. 2:2003(E)、ISO/IEC 9075-9:2001/Cor. 1:2003(E)、ISO/IEC 9075-10:2000/Cor. 1:2003(E)、ISO/IEC 9075-1:1999/Amd. 1:2001/Cor. 1:2003(E)、ISO/IEC 9075-2:1999/Amd. 1:2001/Cor. 1:2003(E) 和 ISO/IEC 9075-5:1999/Amd. 1:2001/Cor. 1:2003(E)——该勘误表包含对 ISO/IEC 9075-1:1999、ISO/IEC 9075-2:1999、ISO/IEC 9075-3:1999、ISO/IEC 9075-4:1999、ISO/IEC 9075-5:1999、ISO/IEC 9075-9:2001、ISO/IEC 9075-10:2000、ISO/IEC 9075-1:1999/Amd. 1:2001、ISO/IEC 9075-2:1999/Amd. 1:2001 和 ISO/IEC 9075-5:1999/Amd. 1:2001 的更正累积。它完全包含并替换了 ISO/IEC 9075 的所有以前的技术勘误。

这些技术勘误表中所包含的 SQL 语言更正都包括在 ISO/IEC 9075:2003 中。

关于 ISO/IEC 9075 中潜在的新问题或新的缺陷报告,可以用下面的地址联系:

Secretariat, ISO/IEC JTC1/SC32
 American National Standards Institute
 11 West 42nd Street
 New York, NY 10036
 USA

附 录 B
(资料性附录)
实现定义的元素

本附录引用 GB/T 12991 的本部分标识为 SQL 实现定义的那些特征。

术语“实现定义的”用来标识实现之间可能不同、但每个特定实现均应定义的特征。

1) 4.2.2“SQL 代理”

在 SQL 直接调用(见 5.3.3.3)中,引起 SQL 语句执行的 SQL 代理是实现定义的。

2) 4.2.4“SQL 客户模块”

a) 创建或撤销 SQL 客户模块的机制是实现定义的。

b) 如果一个 SQL 客户模块有一个名字,其允许的名字是实现定义的。

3) 4.2.8.1“目录”

a) 创建或撤销目录的机制是实现定义的。

b) 在当前 SQL 会话中,通过〈准备语句〉和〈立即执行语句〉动态准备的〈可准备语句〉执行的默认目录名是实现定义的。

4) 4.2.8.2“SQL 模式”

SQL 模式既可以通过执行 SQL 模式语句来创建和撤销,也可以通过实现定义的机制来创建和撤销。

5) 4.4.3.1“数值类型”

如果算术运算的结果不能用结果类型精确表示,则结果是四舍五入还是截断是实现定义的。

6) 4.4.3.2“字符串类型”

a) 字符串类型的最大长度是实现定义的;

b) 字符大对象类型的最大可变长度是实现定义的;

c) 二进制大对象类型的最大可变长度是实现定义的。

7) 4.6.2.2“理序”

所支持的理序若不是由国家标准或国际标准定义的理序派生的,则是实现定义的。

8) 4.8.1“宿主语言”

a) 当使用 SQL 客户模块绑定方式时,说明 SQL 客户模块的机制是实现定义的。

b) 无论选择什么方式,SQL 语句按实现定义的字符集编写,该字符集称为源语言字符集。

9) 4.10.1“一般例程信息”

外部例程对编译单元的引用和该编译单元之间的绑定时间和方法是实现定义的。

10) 4.11.1“SQL 语句分类”

SQL 实现提供的附加 SQL 语句是 SQL 实现定义的。

11) 5.3.3.3“SQL 直接调用”

在 SQL 直接调用过程中,调用 SQL 语句、产生状态、访问诊断信息以及报告结果的方法都是实现定义的。

12) 6.3.3.3“规则求值顺序”

如果 SQL 调用函数是确定的,且不会修改 SQL 数据,则该函数的调用是非本质的;否则,它是本质的还是非本质的是实现定义的。

附 录 C
(资料性附录)
实现相关的元素

本附录引用 GB/T 12991 的本部分明确声明符合性实现的动作是实现相关的。
 术语“实现相关的”用来标识实现之间可能不同、但无需特定实现说明的特征。

1) 4.6.1“一般 SQL 模式对象信息”

一个 SQL 调用例程的专用名,如果没有明确指定,则是实现相关的。

2) 4.6.2.1“字符集”

对于完全包含在某个 SQL 实现中的那些字符,其编码和形成字符串的方法是实现相关的。

3) 4.7.2“确定性和约束”

若一个例程被声明为确定的,而实际上它不是确定的,则其调用效果是实现相关的。

4) 4.8.1“宿主语言”

对于绑定方式的嵌入式 SQL,根据 SQL 语句生成外部调用过程、将那些外部调用过程集成到 SQL 客户模块、将 SQL 语句替换为由它生成的外部调用过程的调用,这些机制都是实现相关的。

5) 6.3.2“信息模式和定义模式的说明”

信息模式视图所基于的实际对象是实现相关的。

6) 6.3.3.2“表示规则要求的术语”

对不符合 GB/T 12991 的格式和句法规则的语言的处理是实现相关的。

7) 6.3.3.7“异常”

a) 除非 GB/T 12991 有明确定义,否则以异常状态终止的 SQL 语句对任何赋值目标和 SQL 描述符区的影响是实现相关的。

b) 如果因一个语句执行出现多个状态,则有关多个状态的诊断信息是否有效是实现相关的。

8) 6.3.3.3“规则求值顺序”

a) 当优先级不是通过格式或括号确定时,表达式实际上是否从左到右求值是实现相关的。

b) 如果一个表达式或搜索条件的非本质部分的求值可能会产生一个异常状态,则是否产生该状态是实现相关的。

c) 包含中间结果的场所的声明类型是实现相关的。

附 录 NA
(资料性附录)

GB/T 12991 标准发展历程

GB/T 12991 经历了两个发展阶段。1991年8月19日我国发布了 GB/T 12991—1991《信息处理系统 数据库语言 SQL》，该标准等同采用国际标准 ISO/IEC 9075—1989《信息处理系统 数据库语言具有完整性增强特征的 SQL》。这是 GB/T 12991 发展的第一阶段。

从 1987 年至今，数据库语言 SQL 国际标准经历了多次修订，其版本有 1987 年版、1989 年版、1992 年版、1999 年版和 2003 年版，从 1999 年版开始将一个标准分成 9 个部分进行规定。

2006 年按照国家标准化管理委员会的国家标准修订计划，全国信息技术标准化技术委员会实施了 GB/T 12991—1991 版的修订工作。GB/T 12991 新版标准等同采用 ISO/IEC 9075 新版系列标准，也将分为 9 个部分，共同替代 GB/T 12991—1991。GB/T 12991 新版标准与 GB/T 12991—1991 相比无论在内容、技术还是在结构上和上一版本相比都作了非常大的调整。这是 GB/T 12991 发展的第二个阶段。

两个版本的具体差异如下：

- 标准的名称发生改变，GB/T 12991—1991 的名称是《信息处理系统 数据库语言 SQL》，而新版名称改为《信息技术 数据库语言 SQL》
- GB/T 12991—1991 是一个独立标准，内容涉及概念、公用元素、模式定义语言、模块语言、数据操纵语言和级 6 个方面内容。而 GB/T 12991 新版是一个拥有 2000 多页内容的系列标准，共分为 9 部分。
- GB/T 12991—1991 标准中的一些术语在新版中都被删除掉了，如：集、数据库、参数、SQL-CODE 参数等。有些术语虽然仍存在，但解释也不一样了，例如：数据类型、列、行、指示符参数等等。
- GB 12991 新版标准中增加了很多内容，例如第 3 部分调用层接口、第 4 部分持久存储模块、第 9 部分外部数据管理、第 10 部分对象语言绑定、第 11 部分信息和定义模式、第 13 部分 Java 历程和类型、第 14 部分 XML 相关的规范等。

参 考 文 献

- [1] [Unicode3.0]The Unicode Consortium, The Unicode Standard, Version 3.0, Reading, MA, Addison—Wesley Developers Press, 2000. ISBN 0-202-61633-5.
- [2] [Unicode 3.1]The Unicode Consortium, The Unicode Standard, Version 3.1.0, Unicode Standard Annex # 27: Unicode 3.1 (which amends The Unicode Standard, Version 3.0). 2001-03-23.
<http://www.unicode.org/unicode/reports/tr27>
- [3] [Unicode 10]Davis, Mark and Whistler, Ken. Unicode Technical Standard # 10, Unicode Collation Algorithm, Version 8.0, 2001-03-23. The Unicode Consortium.
<http://www.unicode.org/unicode/reports/tr10/tr10-8.html>
- [4] [Unicode 15]Davis, Mark and Durst, Martin. Unicode Standard Annex # 15, Unicode Normalization Forms, Version 21.0, 2001-03-23. The Unicode Consortium.
<http://www.unicode.org/unicode/reports/tr15/tr15-21.html>
- [5] [Unicode 19]Davis, Mark. Unicode Standard Annex # 19, UTF-32, Version 8.0, 2001-03-23. The Unicode Consortium.
<http://www.unicode.org/unicode/reports/tr19/tr19-8.html>
-

中 华 人 民 共 和 国
国 家 标 准
信 息 技 术 数 据 库 语 言 SQL
第 1 部 分 : 框 架

GB/T 12991.1—2008/ISO/IEC 9075-1:2003

*

中国标准出版社出版发行
北京复兴门外三里河北街16号
邮政编码:100045

网址 www.spc.net.cn

电话:68523946 68517548

中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

*

开本 880×1230 1/16 印张 3 字数 83 千字
2008年11月第一版 2008年11月第一次印刷

*

书号: 155066·1-34374

如有印装差错 由本社发行中心调换
版权专有 侵权必究
举报电话:(010)68533533



GB/T 12991.1-2008