Atitit sql的标准化与新特性

目录

[1. SQL标准分为十部分。 2](#_Toc27886)

[1.1. 第1部分：](#_Toc30624)*[框架](#_Toc30624)*[（SQL /框架）。它提供了逻辑概念。[41] 2](#_Toc30624)

[1.2. 第2部分：](#_Toc16726)*[基础](#_Toc16726)*[（SQL / Foundation）。它包含语言的最主要元素，并且包含](#_Toc16726)*[强制性和可选性](#_Toc16726)*[功能。 2](#_Toc16726)

[1.3. 第3部分：](#_Toc16195)*[呼叫层介面](#_Toc16195)*[（SQL / CLI）。它定义了接口组件（结构，过程，变量绑定），可用于执行以应用程序中的SQL语句 3](#_Toc16195)

[1.4. 第4部分：](#_Toc11010)*[持久存储的模块](#_Toc11010)*[（SQL / PSM） sp udf 3](#_Toc11010)

[1.5. 第9部分：](#_Toc4896)*[外部数据管理](#_Toc4896)*[（SQL / MED）。 3](#_Toc4896)

[1.6. 第10部分：](#_Toc1246)*[对象语言绑定](#_Toc1246)*[（SQL / OLB）。它定义了SQLJ的语法和语义，SQLJ是Java中嵌入的SQL（另请参见第3部分） 3](#_Toc1246)

[1.7. 第11部分：](#_Toc27852)*[信息和定义架构](#_Toc27852)*[（SQL / Schemata）。 4](#_Toc27852)

[1.8. 第13部分：](#_Toc2876)*[使用Java TM编程语言](#_Toc2876)*[（SQL / JRT） 4](#_Toc2876)

[1.9. 第14部分：与](#_Toc17459)*[XML相关的规范](#_Toc17459)*[（SQL / XML）。 4](#_Toc17459)

[1.10. 第15部分：](#_Toc22446)*[多维数组](#_Toc22446)*[（SQL / MDA）。 5](#_Toc22446)

[2. 废弃的步伐 5](#_Toc12425)

[3. Sql标准化的历史 5](#_Toc16578)

[3.1. 1986年 SQL-86 SQL-87 首先由ANSI规范化。 6](#_Toc431)

[3.2. 1989年 SQL-89 ，增加了完整性约束 6](#_Toc25676)

[3.3. Sql92标准 7](#_Toc7604)

[3.3.1. 定义了新的数据类型：DATE，TIME，TIMESTAMP， 8](#_Toc6511)

[3.3.2. 新的标量运算，例如字符串连接和子字符串提取，日期和时间数学以及条件语句。 8](#_Toc2904)

[3.3.3. 1996年的SQL / PSM（存储过程） 9](#_Toc12163)

[3.4. Sql99标准 9](#_Toc6591)

[3.4.1. 使用Java编程语言存储过程 10](#_Toc18819)

[3.4.2. 触发器，对过程和流程控制语句的支持 10](#_Toc22204)

[3.4.3. ，非标量类型（数组）和一些面向对象的功能（例如，结构化类型）。 10](#_Toc25013)

[3.4.4. 支持在Java（SQL / OLB）中嵌入SQL，反之亦然（SQL / JRT）。 10](#_Toc20561)

[3.5. SQL:2003为例，它包括以下9个部分 10](#_Toc8089)

[3.5.1. 与XML相关的功能（SQL / XML） 11](#_Toc17243)

[3.5.2. 窗口功能 11](#_Toc14962)

[3.5.3. 两种新的列类型：自动生成的值和标识列 11](#_Toc10379)

[3.5.4. 新的MERGE语句 11](#_Toc13530)

[3.5.5. 对CREATE TABLE语句的扩展，以允许“ CREATE TABLE AS”和“ CREATE TABLE LIKE” 11](#_Toc23051)

[3.6. Sql2006标准 11](#_Toc25160)

[3.6.1. 可以使用XQuery 11](#_Toc8079)

[3.7. Sql2008标准 12](#_Toc28017)

[3.7.1. TRUNCATETABLE语句， 12](#_Toc7067)

[3.7.2. FETCH子句 12](#_Toc3805)

[3.7.3. INSTEAD 数据库触发器 12](#_Toc11331)

[3.7.4. 分区连接表， 12](#_Toc23870)

[3.7.5. 各种XQuery的正则表达式/模式匹配功能的支持，并 13](#_Toc8887)

[3.8. SQL:2011 13](#_Toc11580)

[3.9. 2016年 SQL：2016年 添加行模式匹配，多态表函数JSON。 14](#_Toc4178)

[3.10. 2019年 SQL：2019年 添加了第15部分，多维数组（MDarray类型和运算符）。 14](#_Toc25807)

# SQL标准分为十部分。

## 第1部分：*框架*（SQL /框架）。它提供了逻辑概念。[[41]](https://en.wikipedia.org/wiki/SQL" \l "cite_note-ISO/IEC-44)

ISO / IEC 9075-1：2016

## 第2部分：*基础*（SQL / Foundation）。它包含语言的最主要元素，并且包含*强制性和可选性*功能。

ISO / IEC 9075-2：2016

## 第3部分：*呼叫层介面*（[SQL / CLI](https://en.wikipedia.org/wiki/SQL/CLI" \o "SQL / CLI)）。它定义了接口组件（结构，过程，变量绑定），可用于执行以应用程序中的SQL语句

ISO / IEC 9075-3：2016

。（对于Java，请参阅第10部分。）以这样一种方式定义SQL / CLI：将SQL语句和SQL / CLI过程调用与调用应用程序的源代码分开对待。[开放式数据库连接](https://en.wikipedia.org/wiki/Open_Database_Connectivity" \o "开放式数据库连接)是SQL / CLI的著名超集。标准的这一部分仅包含*强制性*功能。

## 第4部分：*持久存储的模块*（[SQL / PSM](https://en.wikipedia.org/wiki/SQL/PSM" \o "SQL / PSM)） sp udf

ISO / IEC 9075-4：2016

* 。它标准化了SQL的过程扩展，包括控制流，条件处理，语句条件信号和信号，游标和局部变量，以及将表达式分配给变量和参数。另外，SQL / PSM使持久数据库语言例程（例如“存储过程”）的声明和维护形式化。标准的这一部分仅包含*可选*功能。

## 第9部分：*外部数据管理*（[SQL / MED](https://en.wikipedia.org/wiki/SQL/MED" \o "SQL / MED)）。

* 它提供了SQL扩展，用于定义外部数据包装器和数据链接类型，以允许SQL管理外部数据。外部数据是基于SQL的DBMS可以访问但不能由其管理的数据。标准的这一部分仅包含*可选*功能。
* ISO / IEC 9075-9：2016

## 第10部分：*对象语言绑定*（[SQL / OLB](https://en.wikipedia.org/wiki/SQL/OLB" \o "SQL / OLB)）。它定义了[SQLJ](https://en.wikipedia.org/wiki/SQLJ" \o "SQLJ)的语法和语义，[SQLJ](https://en.wikipedia.org/wiki/SQLJ" \o "SQLJ)是Java中嵌入的SQL（另请参见第3部分）

ISO / IEC 9075-10：2016

* 。该标准还描述了确保SQLJ应用程序二进制可移植性的机制，并指定了各种Java软件包及其包含的类。标准的这一部分仅包含*可选*功能。与SQL / OLB不同，[JDBC](https://en.wikipedia.org/wiki/JDBC" \o "JDBC)定义了[API](https://en.wikipedia.org/wiki/API" \o "API)，并且不是SQL标准的一部分。[ *[需要引用](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "维基百科：需要引用)* ]

## 第11部分：*信息和定义架构*（[SQL / Schemata](https://en.wikipedia.org/wiki/SQL/Schemata" \o "SQL /图式)）。

* 它定义了信息模式和定义模式，提供了一组通用的工具来使SQL数据库和对象自描述。这些工具包括SQL对象标识符，结构和完整性约束，安全性和授权规范，ISO / IEC 9075的功能和软件包，对基于SQL的DBMS实施提供的功能的支持，基于SQL的DBMS实施信息和大小调整项以及DBMS实现支持的值。[[42]](https://en.wikipedia.org/wiki/SQL" \l "cite_note-ISO/IEC_9075-11:2008-45)标准的这一部分包含*强制性和可选性*。
* ISO / IEC 9075-11：2016

## 第13部分：*使用Java TM编程语言*（[SQL / JRT](https://en.wikipedia.org/wiki/SQL/JRT" \o "SQL / JRT)）

* 的*SQL例程和类型*。它指定了从SQL应用程序（“数据库中的Java”）中将静态Java方法作为例程调用的功能。它还要求能够将Java类用作SQL结构化的用户定义类型。标准的这一部分仅包含*可选*功能。
* ISO / IEC 9075-13：2016

## 第14部分：与*XML相关的规范*（[SQL / XML](https://en.wikipedia.org/wiki/SQL/XML" \o "SQL / XML)）。

* 它指定了将SQL与SQL结合使用的基于SQL的扩展。的*XML*数据类型被引入，以及几个例程，函数，和XML到SQL数据类型映射到支持操作和XML的存储在SQL数据库中。[[35]](https://en.wikipedia.org/wiki/SQL" \l "cite_note-SQLXML2006-38)标准的这一部分仅包含*可选*功能。[ *[需要引用](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "维基百科：需要引用)* ]
* ISO / IEC 9075-14：2016

## 第15部分：*多维数组*（SQL / MDA）。

它为SQL指定了多维数组类型（MDarray），

以及对MDarray，MDarray切片，MDarray单元和相关功能的操作。标准的这一部分仅包含*可选*功能。

ISO / IEC 9075-15：2019

# 废弃的步伐

ISO / IEC 9075-5：1999

信息技术—数据库语言— SQL —第5部分：主机语言绑定（SQL /绑定）

ISO / IEC 9075

维基百科，自由的百科全书

跳转到导航跳转到搜索

ISO / IEC 9075 标准：“信息技术-数据库语言-SQL”，它描述了结构化查询语言。

查看发布的详细信息：

SQL-86（或SQL-87）是1987年的ISO 9075：1987标准

SQL-89是1989年的ISO / IEC 9075：1989标准

SQL-92是1992年的ISO / IEC 9075：1992标准

SQL：1999是1999年的ISO / IEC 9075：1999标准

SQL：2003是2003的ISO / IEC 9075：2003标准

SQL：2006是2006年的ISO / IEC 9075：2006标准

SQL：2008是2008年的ISO / IEC 9075：2008标准

SQL：2011是2011年的ISO / IEC 9075：2011标准

SQL：2016是2016年的ISO / IEC 9075：2016标准

# Sql标准化的历史

在20世纪60年代，网状数据库系统（如CODASYL）和分层数据库系统（如IMS TM）是用于自动化银行业务、记帐和订单处理系统的一流技术，这些系统是由于商业大型计算机的引入才启用的。而SQL是在70年代创建的一种基于关系数据库管理系统（Relational Database Management System，RDBMS）模型的数据查询、操作语言

SQL是一种关系型数据库查询语言，要介绍SQL的起源，就不得不介绍IBM公司的两个重量级人物—E.F. Codd博士和Don Chamberlin博士。E. F. Codd博士最早提出了关系数据库管理系统（Relational Database Management System，RDBMS）模型，而Don Chamberlin博士则是SQL和XQuery语言的主要创造者之一。他们对数据库的变革起到了革命性的作用

虽然IBM首创了关系数据库理论，但Oracle却是第一家在市场上推出了这套技术的公司。随着时间的推移，SQL的简洁、直观，在市场上获得了的不错反响，从而引起了ANSI的关注，分别在1986年、1989年、1992年、1999年及2003年发布了SQL标准。SQL Server 2000遵循ANSI SQL:1992标准，而SQL Server 2005和2008还实现了ANSI SQL:1999和ANSI SQL:2003中的一些重要特性。

对SQL标准影响最大的机构自然是那些著名的数据库产商，而具体的制订者则是一些非营利机构，例如国际标准化组织ISO、美国国家标准委员会ANSI等

　下面是SQL发展的简要历史：  
1986年，ANSI X3.135-1986，ISO/IEC 9075:1986，SQL-86   
1989年，ANSI X3.135-1989，ISO/IEC 9075:1989，SQL-89   
1992年，ANSI X3.135-1992，ISO/IEC 9075:1992，SQL-92（SQL2）

比较推荐的方法是泛读SQL92（因为它涉及了SQL最基础和最核心的一些内容  
1999年，ISO/IEC 9075:1999，SQL:1999（SQL3）

并于1999年再次更新为SQL99和SQL3标准。在每一次更新中，ANSI都在SQL中添加了新特性，并在语言中集成了新的命令和功能。  
2003年，ISO/IEC 9075:2003，SQL:2003  
2008年，ISO/IEC 9075:2008，SQL:2008  
2011年，ISO/IEC 9075:2011，SQL:2011

SQL86大概只有几十页，SQL92正文大约有500页，而SQL99则超过了1000页。可以看出，从SQL99开始，SQL标准的个头就非常庞大了，内容包罗万象，已经没有人能够掌握标准的所有内容了

## 1986年 SQL-86 SQL-87 首先由ANSI规范化。

## 1989年 SQL-89 ，增加了完整性约束

1999年 SQL：1999 SQL3 添加了正则表达式匹配，递归查询（例如，传递闭包），触发器，对过程和流程控制语句的支持，非标量类型（数组）和一些面向对象的功能（例如，结构化类型）。支持在Java（SQL / OLB）中嵌入SQL，反之亦然（SQL / JRT）。

2003年 SQL：2003 引入了与XML相关的功能（SQL / XML），窗口函数，标准化序列以及具有自动生成的值（包括标识列）的列。

2006年 SQL：2006 ISO / IEC 9075-14：2006定义了SQL与XML结合使用的方式。它定义了在SQL数据库中导入和存储XML数据，在数据库中进行操作以及以XML形式发布XML和常规SQL数据的方法。此外，它还使应用程序可以使用XQuery（万维网联盟（W3C）发布的XML查询语言）将查询集成到SQL代码中，以同时访问普通的SQL数据和XML文档。[35]

2008年 SQL：2008 使外部游标定义中的ORDER BY合法化。添加INSTEAD OF触发器，TRUNCATE语句，[36] FETCH子句。

2011年 SQL：2011年 添加时态数据（PERIOD FOR）[37]（更多信息，请访问：时态数据库#History）。窗口功能和FETCH子句的增强。[38]

2016年 SQL：2016年 添加行模式匹配，多态表函数JSON。

2019年 SQL：2019年 添加了第15部分，多维数组（MDarray类型和运算符）。

## Sql92标准

New features

Significant new features include:[[2]](https://en.wikipedia.org/wiki/SQL-92" \l "cite_note-2)

* New data types defined: DATE, TIME, TIMESTAMP, INTERVAL, BIT string, VARCHAR strings, and NATIONAL CHARACTER strings.
* Support for additional [character sets](https://en.wikipedia.org/wiki/Character_set" \o "Character set) beyond the base requirement for representing SQL statements.
* New scalar operations such as string concatenation and [substring](https://en.wikipedia.org/wiki/Substring" \o "Substring) extraction, date and time mathematics, and conditional statements.
* New set operations such as UNION JOIN, NATURAL JOIN, set differences, and set intersections.
* Conditional expressions with CASE. For an example, see [Case (SQL)](https://en.wikipedia.org/wiki/Case_(SQL)" \o "Case (SQL)).
* Support for alterations of [schema definitions](https://en.wikipedia.org/wiki/Data_Definition_Language" \o "Data Definition Language) via ALTER and DROP.
* Bindings for [C](https://en.wikipedia.org/wiki/C_(programming_language)" \o "C (programming language)), [Ada](https://en.wikipedia.org/wiki/Ada_(programming_language)" \o "Ada (programming language)), and [MUMPS](https://en.wikipedia.org/wiki/MUMPS" \o "MUMPS).
* New features for user privileges.
* New integrity-checking functionality such as within a [CHECK constraint](https://en.wikipedia.org/wiki/Check_Constraint" \o "Check Constraint).
* A new *information schema*—read-only views about database metadata like what tables it contains, etc. For example, SELECT \* FROM INFORMATION\_SCHEMA.TABLES;.
* Dynamic execution of queries (as opposed to prepared).
* Better support for remote database access.
* Temporary tables; CREATE TEMP TABLE etc.
* [Transaction](https://en.wikipedia.org/wiki/Database_transaction" \o "Database transaction) [isolation levels](https://en.wikipedia.org/wiki/Isolation_(database_systems)" \o "Isolation (database systems)).
* New operations for changing data types on the fly via CAST (expr AS type).
* [Scrolled cursors](https://en.wikipedia.org/wiki/Cursor_(databases)" \l "Scrollable_cursors" \o "Cursor (databases)).
* Compatibility flagging for backwards and forwards compatibility with other SQL standards.

Extensions

Two significant extensions were published after standard (but before the next major iteration.)

* [SQL/CLI](https://en.wikipedia.org/wiki/SQL/CLI" \o "SQL/CLI) ([Call Level Interface](https://en.wikipedia.org/wiki/Call_Level_Interface" \o "Call Level Interface)) in 1995
* [SQL/PSM](https://en.wikipedia.org/wiki/SQL/PSM" \o "SQL/PSM) ([stored procedures](https://en.wikipedia.org/wiki/Stored_procedure" \o "Stored procedure)) in 1996

重要的新功能包括：[2]

### 定义了新的数据类型：DATE，TIME，TIMESTAMP，

INTERVAL，BIT字符串，VARCHAR字符串和NATIONAL CHARACTER字符串。

支持超出代表SQL语句的基本要求的其他字符集。

### 新的标量运算，例如字符串连接和子字符串提取，日期和时间数学以及条件语句。

新的设置操作，例如UNION JOIN，NATURAL JOIN，设置差异和设置相交。

使用CASE的条件表达式。有关示例，请参见Case（SQL）。

支持通过ALTER和DROP更改架构定义

新的信息模式-有关数据库元数据的只读视图，例如数据库包含的表等。例如，SELECT \* FROM INFORMATION\_SCHEMA.TABLES;。

动态执行查询（与准备执行相反）。

更好地支持远程数据库访问。

临时表；创建温度表等

事务隔离级别。

通过CAST（expr AS类型）即时更改数据类型的新操作。

滚动的光标。

兼容性标记，用于与其他SQL标准向后和向前兼容。

扩展名

在标准之后（但在下一个主要迭代之前）发布了两个重要的扩展。

1995年的SQL / CLI（呼叫层接口）

### 1996年的SQL / PSM（存储过程）

## Sql99标准

· [2 New features](https://en.wikipedia.org/wiki/SQL:1999" \l "New_features)

* [2.1 Data types](https://en.wikipedia.org/wiki/SQL:1999" \l "Data_types)
  + [2.1.1 Boolean data types](https://en.wikipedia.org/wiki/SQL:1999" \l "Boolean_data_types)
  + [2.1.2 Distinct user-defined types of power](https://en.wikipedia.org/wiki/SQL:1999" \l "Distinct_user-defined_types_of_power)
  + [2.1.3 Structured user-defined types](https://en.wikipedia.org/wiki/SQL:1999" \l "Structured_user-defined_types)
* [2.2 Common table expressions and recursive queries](https://en.wikipedia.org/wiki/SQL:1999" \l "Common_table_expressions_and_recursive_queries)
* [2.3 Some OLAP capabilities](https://en.wikipedia.org/wiki/SQL:1999" \l "Some_OLAP_capabilities)
* [2.4 Role-based access control](https://en.wikipedia.org/wiki/SQL:1999" \l "Role-based_access_control)

Summary

The ISO standard documents were published between 1999 and 2002 in several installments, the first one consisting of multiple parts. Unlike previous editions, the standard's name used a colon instead of a hyphen for consistency with the names of other [ISO](https://en.wikipedia.org/wiki/International_Organization_for_Standardization" \o "International Organization for Standardization) standards. The first installment of SQL:1999 had five parts:

* [SQL/Framework](https://en.wikipedia.org/w/index.php?title=SQL/Framework&action=edit&redlink=1" \o "SQL/Framework (page does not exist)) [ISO/IEC 9075-1:1999](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26196)
* [SQL/Foundation](https://en.wikipedia.org/w/index.php?title=SQL/Foundation&action=edit&redlink=1" \o "SQL/Foundation (page does not exist)) [ISO/IEC 9075-2:1999](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26197)
* [SQL/CLI](https://en.wikipedia.org/wiki/SQL/CLI" \o "SQL/CLI) : an updated definition of the extension Call Level Interface, originally published in 1995, also known as CLI-95 [ISO/IEC 9075-3:1999](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=30609)
* [SQL/PSM](https://en.wikipedia.org/wiki/SQL/PSM" \o "SQL/PSM) : an updated definition of the extension Persistent Stored Modules, originally published in 1996, also known as PSM-96 [ISO/IEC 9075-4:1999](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=29864)
* [SQL/Bindings](https://en.wikipedia.org/w/index.php?title=SQL/Bindings&action=edit&redlink=1" \o "SQL/Bindings (page does not exist)) [ISO/IEC 9075-5:1999](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26198)

Three more parts, also considered part of SQL:1999 were published subsequently:

* [SQL/MED](https://en.wikipedia.org/wiki/SQL/MED" \o "SQL/MED) Management of External Data (SQL:1999 part 9) [ISO/IEC 9075-9:2001](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=31370)
* [SQL/OLB](https://en.wikipedia.org/wiki/SQL/OLB" \o "SQL/OLB) Object Language Bindings (SQL:1999 part 10) [ISO/IEC 9075-10:2000](http://www.iso.org/iso/catalogue_detail.htm?csnumber=30613)
* [SQL/JRT](https://en.wikipedia.org/wiki/SQL/JRT" \o "SQL/JRT) SQL Routines and Types using the Java Programming Language (SQL:1999 part 13) [ISO/IEC 9075-13:2002](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=35340)

### 使用Java编程语言存储过程

添加了正则表达式匹配，[递归查询](https://en.wikipedia.org/wiki/Hierarchical_and_recursive_queries_in_SQL" \o "SQL中的分层查询和递归查询)（例如，[传递闭包](https://en.wikipedia.org/wiki/Transitive_closure" \o "传递闭包)），

### [触发器](https://en.wikipedia.org/wiki/Database_trigger" \o "数据库触发器)，对过程和流程控制语句的支持

### ，非标量类型（数组）和一些面向对象的功能（例如，[结构化类型](https://en.wikipedia.org/wiki/Structured_type" \o "结构化类型)）。

### 支持在Java（[SQL / OLB](https://en.wikipedia.org/wiki/SQL/OLB" \o "SQL / OLB)）中嵌入SQL，反之亦然（[SQL / JRT](https://en.wikipedia.org/wiki/SQL/JRT" \o "SQL / JRT)）。

## SQL:2003为例，它包括以下9个部分

（中间编号空缺是曾经被占用，之后被废弃的标准造成的）：

ISO/IEC9075-1: Framework (SQL/Framework)  
ISO/IEC 9075-2: Foundation (SQL/Foundation)  
ISO/IEC 9075-3: Call Level Interface (SQL/CLI)  
ISO/IEC 9075-4: Persistent Stored Modules (SQL/PSM)  
ISO/IEC 9075-9: Management of External Data (SQL/MED)  
ISO/IEC 9075-10: Object Language Bindings (SQL/OLB)  
ISO/IEC 9075-11: Information and Definition Schemas (SQL/Schemata)  
ISO/IEC 9075-13: [Java](http://www.2cto.com/kf/ware/Java/" \t "http://www.2cto.com/database/201412/_blank) Routines and Types Using the Java Programming Language(SQL/JRT)  
ISO/IEC 9075-14: XML-Related Specifications (SQL/XML)

Summary

The SQL:2003 standard makes minor modifications to all parts of [SQL:1999](https://en.wikipedia.org/wiki/SQL:1999" \o "SQL:1999) (also known as SQL3), and officially introduces a few new features such as:[[1]](https://en.wikipedia.org/wiki/SQL:2003" \l "cite_note-1)

* [XML](https://en.wikipedia.org/wiki/XML" \o "XML)-related features ([SQL/XML](https://en.wikipedia.org/wiki/SQL/XML" \o "SQL/XML))
* [Window functions](https://en.wikipedia.org/wiki/Window_function_(SQL)" \o "Window function (SQL))
* the sequence generator, which allows standardized sequences
* two new column types: auto-generated values and identity-columns
* the new [MERGE](https://en.wikipedia.org/wiki/Merge_(SQL)" \o "Merge (SQL)) statement
* extensions to the [CREATE TABLE](https://en.wikipedia.org/wiki/Data_Definition_Language" \o "Data Definition Language) statement, to allow "CREATE TABLE AS" and "CREATE TABLE LIKE"
* removal of the poorly implemented "BIT" and "BIT VARYING" data types
* [OLAP](https://en.wikipedia.org/wiki/OLAP" \o "OLAP) capabilities (initially added in [SQL:1999](https://en.wikipedia.org/wiki/SQL:1999" \o "SQL:1999)) were extended with a [window function](https://en.wikipedia.org/wiki/Window_function_(SQL)" \o "Window function (SQL)).[[2]](https://en.wikipedia.org/wiki/SQL:2003" \l "cite_note-JensenPedersen2010-2)

### 与XML相关的功能（SQL / XML）

### 窗口功能

序列生成器，允许标准化序列

### 两种新的列类型：自动生成的值和标识列

### 新的MERGE语句

### 对CREATE TABLE语句的扩展，以允许“ CREATE TABLE AS”和“ CREATE TABLE LIKE”

删除实施不佳的“ BIT”和“ BIT VARYING”数据类型

OLAP功能（最初在SQL：1999中添加）通过窗口函数进行了扩展。[2]

## Sql2006标准

### 可以使用[XQuery](https://en.wikipedia.org/wiki/XQuery" \o "XQuery)

The main changes from [SQL:2003](https://en.wikipedia.org/wiki/SQL:2003" \o "SQL:2003) were in the Part 14 of the standard.

Part 14

ISO/IEC 9075-14:2006 defines ways in which SQL can be used in conjunction with XML. It defines ways of importing and storing XML data in an SQL database, manipulating it within the database and publishing both XML and conventional SQL-data in XML form. In addition, it enables applications to integrate into their SQL code the use of [XQuery](https://en.wikipedia.org/wiki/XQuery" \o "XQuery), the XML Query Language published by the World Wide Web Consortium ([W3C](https://en.wikipedia.org/wiki/W3C" \o "W3C)), to concurrently access ordinary SQL-data and XML documents.

## Sql2008标准

Sql2008也包括2003继承的那几大部分

增加的包括新特性

* enhanced [MERGE](https://en.wikipedia.org/wiki/Merge_(SQL)" \o "Merge (SQL)) and DIAGNOSTIC statements,
* the [TRUNCATE TABLE](https://en.wikipedia.org/wiki/Truncate_(SQL)" \o "Truncate (SQL)) statement,
* comma-separated WHEN clauses in a CASE expression,
* INSTEAD OF [database triggers](https://en.wikipedia.org/wiki/Database_trigger" \o "Database trigger)
* partitioned [JOIN](https://en.wikipedia.org/wiki/Join_(SQL)" \o "Join (SQL)) tables,
* support for various [XQuery](https://en.wikipedia.org/wiki/XQuery" \o "XQuery) [regular expression](https://en.wikipedia.org/wiki/Regular_expression" \o "Regular expression)/[pattern-matching](https://en.wikipedia.org/wiki/Pattern_matching" \o "Pattern matching) features, and
* enhancements to derived column names.[[1]](https://en.wikipedia.org/wiki/SQL:2008" \l "cite_note-sybase08-1)

增强合并和诊断 ，

### TRUNCATETABLE语句，

用逗号分隔的表达式

### FETCH子句

### INSTEAD 数据库触发器

### 分区连接表，

### 各种XQuery的正则表达式/模式匹配功能的支持，并

增强派生列名称。[ 1 ]

XML相关规范定义了如何在SQL可以配合使用XML，包括进口和在SQL数据库中存储XML数据，操纵它在数据库和发布XML和XML格式的传统SQL数据。[ 2 ]

## SQL:2011

New features

Temporal support

One of the main new features is improved support for [temporal databases](https://en.wikipedia.org/wiki/Temporal_database" \o "Temporal database).[[3]](https://en.wikipedia.org/wiki/SQL:2011" \l "cite_note-3)[[4]](https://en.wikipedia.org/wiki/SQL:2011" \l "cite_note-4) Language enhancements for temporal data definition and manipulation include:

* **Time Period definitions** use two standard table columns as the start and end of a named time period, with closed-open semantics. This provides compatibility with existing data models, application code, and tools
* Definition of **application time period tables** (elsewhere called [valid time](https://en.wikipedia.org/wiki/Valid_time" \o "Valid time) tables), using the PERIOD FOR annotation
* Update and deletion of application time rows with **automatic time period splitting**
* **Temporal [primary keys](https://en.wikipedia.org/wiki/Primary_key" \o "Primary key)** incorporating application time periods with optional non-overlapping constraints via the WITHOUT OVERLAPS clause
* **Temporal [referential integrity](https://en.wikipedia.org/wiki/Referential_integrity" \o "Referential integrity)** constraints for application time tables
* Application time tables are queried using regular query syntax or using new **temporal predicates** for time periods including CONTAINS, OVERLAPS, EQUALS, PRECEDES, SUCCEEDS, IMMEDIATELY PRECEDES, and IMMEDIATELY SUCCEEDS (which are modified versions of [Allen’s interval relations](https://en.wikipedia.org/wiki/Allen's_interval_algebra" \l "Relations" \o "Allen's interval algebra))
* Definition of **system-versioned tables** (elsewhere called [transaction time](https://en.wikipedia.org/wiki/Transaction_time" \o "Transaction time) tables), using the PERIOD FOR SYSTEM\_TIME annotation and WITH SYSTEM VERSIONING modifier. System time periods are maintained automatically. Constraints for system-versioned tables are not required to be temporal and are only enforced on current rows
* Syntax for **time-sliced** and **sequenced** queries on system time tables via the AS OF SYSTEM TIME and VERSIONS BETWEEN SYSTEM TIME ... AND ... clauses
* Application time and system versioning can be used together to provide **[bitemporal](https://en.wikipedia.org/wiki/Bitemporal" \o "Bitemporal) tables**

[IBM DB2](https://en.wikipedia.org/wiki/IBM_DB2" \o "IBM DB2) version 10 claims to be the first database to have a conforming implementation of this feature in what they call "Time Travel Queries".,[[5]](https://en.wikipedia.org/wiki/SQL:2011" \l "cite_note-5)[[6]](https://en.wikipedia.org/wiki/SQL:2011" \l "cite_note-6) although they use the alternative syntax FOR SYSTEM\_TIME AS OF.

[Oracle](https://en.wikipedia.org/wiki/Oracle_Database" \o "Oracle Database) Oracle 12c supports temporal functionality in compliance with SQL:2011.[[7]](https://en.wikipedia.org/wiki/SQL:2011" \l "cite_note-7) Versions 10g and 11g implement the **time-sliced** queries in what they call **Flashback Queries**, using the alternative syntax AS OF TIMESTAMP.[[8]](https://en.wikipedia.org/wiki/SQL:2011" \l "cite_note-8) Notably both of Oracle's implementations depend on the database [transaction log](https://en.wikipedia.org/wiki/Transaction_log" \o "Transaction log) and so only allow temporal queries against recent changes which are still being retained for backup.

[Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server" \o "Microsoft SQL Server) (version 2016) implements temporal tables with SYSTEM\_VERSIONING.[[9]](https://en.wikipedia.org/wiki/SQL:2011" \l "cite_note-9)

临时支持

主要的新功能之一是改进了对时态数据库的支持。[3] [4]时间数据定义和操作的语言增强包括：

## 2016年 SQL：2016年 添加行模式匹配，多态表函数JSON。

## 2019年 SQL：2019年 添加了第15部分，多维数组（MDarray类型和运算符）。

Ref

SQL-维基百科