Atitit。sql2016标准化的规划方案 v3 q2a

[1. Sql标准化的历史 3](#_Toc2298)

[1.1. Sql92标准 4](#_Toc19546)

[1.2. Sql99标准 4](#_Toc5824)

[1.3. SQL:2003为例，它包括以下9个部分 5](#_Toc6575)

[1.4. Sql2006标准 6](#_Toc19304)

[1.5. Sql2008标准 7](#_Toc9181)

[1.6. SQL:2011 7](#_Toc14864)

[2. 多媒体大文件数据特点 8](#_Toc8050)

[3. 现有数据存储（包括但不限于数据库）的问题 9](#_Toc9038)

[3.1. 数据存储查询没有统一标准在文件，内存数据结构，cache，网络数据，与数据库，需要统一的访问模式 9](#_Toc14309)

[3.2. 二维数据结构，对多维数据 层次数据表达能力比较差 9](#_Toc9485)

[3.3. 二进制模式存储，不便于用户使用简单工具直接修改 9](#_Toc31362)

[3.4. 未有集成的可以简单的自定义的gui界面，现有gui界面不可配置化 9](#_Toc2636)

[3.5. 对数据的编辑不能根据类型自动调用不同的编辑控件 9](#_Toc13730)

[3.6. 对oo对象数据的存储支持不足 9](#_Toc16880)

[3.7. 对多媒体 大文件数据的存储支持不足 9](#_Toc22211)

[3.8. Sql表达能力不足，上下文无关 9](#_Toc32005)

[3.9. 插件化不足 9](#_Toc26169)

[4. 网上搜集的到的资料--存储的未来趋势 9](#_Toc14370)

[5. 预计规划sql2016 attilax版的实现目标 10](#_Toc30093)

[5.1. 无模式（schema-free no——schema 10](#_Toc20681)

[5.2. 大力提升用户体验 文本化 透明化 可视化数据管理 10](#_Toc15822)

[5.3. Cms子系统内容管理增强 10](#_Toc3839)

[5.4. 多媒体数据与大文件数据直接支持 多维数据的支持 10](#_Toc21850)

[5.5. Oo化 LINQ 11](#_Toc3186)

[5.6. Fulltxt全文索引 与摘要索引 11](#_Toc24802)

[5.7. 微服务soa化 11](#_Toc28123)

[5.8. Other 11](#_Toc450)

[5.8.1. 自定义类型与类型值读取器 12](#_Toc18571)

[5.8.2. 导入导出 12](#_Toc5088)

[5.8.3. 可视化与报表 12](#_Toc7490)

[5.8.4. Uml模型 12](#_Toc8106)

[5.8.5. Web查询与更新编辑支持（这个好像已经sql2008有了，只需要实现了） 12](#_Toc23224)

[5.8.6. Pagging翻页 12](#_Toc18674)

[5.8.7. INFORMATION\_SCHEMA文件夹模式物理存储 12](#_Toc11652)

[5.8.8. 驱动api标准化 12](#_Toc2488)

[5.8.9. ftp smb nfs支持，方便用户远程编辑记录 12](#_Toc20561)

[5.8.10. ETL工具 13](#_Toc16409)

[5.8.11. AS SERVICE 13](#_Toc26225)

[5.8.12. Logic store逻辑as数据 13](#_Toc4362)

[6. 参考 13](#_Toc31825)

# 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

## 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

## 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)

## 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)

## Sql2006标准

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

代替数据库触发器

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)

# 多媒体大文件数据特点

数据条数可能不是很多，几万足以。。

但是数据体积很庞大,高达30T..主要是视频文件字段超大。。

为了方便数据维护，需要无专有工具模式维护。。

# 现有数据存储（包括但不限于数据库）的问题

## 数据存储查询没有统一标准在文件，内存数据结构，cache，网络数据，与数据库，需要统一的访问模式

## 二维数据结构，对多维数据 层次数据表达能力比较差

## 二进制模式存储，不便于用户使用简单工具直接修改

## 未有集成的可以简单的自定义的gui界面，现有gui界面不可配置化

## 对数据的编辑不能根据类型自动调用不同的编辑控件

## 对oo对象数据的存储支持不足

## 对多媒体 大文件数据的存储支持不足

## Sql表达能力不足，上下文无关

## 插件化不足

# 网上搜集的到的资料--存储的未来趋势

数据挖掘Data Mining)，大数据化，AS SERVICE

、联机分析处理（OLAP）

数据仓库

智能商务运算bi、

集成化

NoSQL将回归SQL

NoSQL将来的归宿就是SQL， Cassandra和Mongo都发布了高级语言，其实就是SQL。

纯低级语言的价值已经随风而去，NoSQL系统将拥抱ACID兼容规范，

# 预计规划sql2016 attilax版的实现目标

## 无模式（schema-free no——schema

## 大力提升用户体验 文本化 透明化 可视化数据管理

透明的物理存储结构，使用文件夹+文本文件+二进制文件，方便小白用户维护数据记录以及文件

小白用户数据维护计划，通用gui工具（尽可能使用文件夹资源管理器+记事本+excel 来维护数据，方便修改）

文本化的数据记录，方便维护

数据记录根据情况使用json yaml csv格式存储

文本化也是比较复合趋势，理解第一，维护方便第一。性能第二。

现在许多协议也使用文本化的了，比如ssdp upnp就直接建立在http上，抛弃了二进制的实现模式

文本化的索引结构物理存储，方便维护查看检测等。。

具体存储可采用文件夹模式，或者半文本化，采用复合文档、注册表来存储

## Cms子系统内容管理增强

无后台管理数据，目前貌似好多程序目的就是为了管理数据而产生的，可以砍掉或大大简化后台管理功能了

使用通用gui工具（尽可能的可以使用文件夹资源管理器+记事本+excel 来维护数据，方便修改）

## 多媒体数据与大文件数据直接支持 多维数据的支持

内建的media数据类型支持。

文档型数据word，pdf，html支持

压缩文件支持 rar zip

其他二进制文件

对网状数据，层次化子表，oo数据存储的实现

## Oo化 LINQ

API内部使用mysql兼容，外部linq化

Json化sql实现

全面通用性语言支持

现有的sql是有选择 循环语句的，但是貌似只能存储过程中使用。。

准备全面实现上下文相关的语言特性支持。这样就比较完整了。

直接支持编程语言函数（java c# php js）计划

  LINQ的优点

    LINQ的主要优点体现在它是一种标准，不但可以在一个[关系数据库](http://cpro.baidu.com/cpro/ui/uijs.php?adclass=0&app_id=0&c=news&cf=1001&ch=0&di=128&fv=20&is_app=0&jk=bfa985f20636b36f&k=%B9%D8%CF%B5%CA%FD%BE%DD%BF%E2&k0=%B9%D8%CF%B5%CA%FD%BE%DD%BF%E2&kdi0=0&luki=2&mcpm=0&n=10&p=baidu&q=csai_cpr&rb=0&rs=1&seller_id=1&sid=6fb33606f285a9bf&ssp2=1&stid=9&t=tpclicked3_hc&td=1730417&tu=u1730417&u=http://www.shangxueba.com/jingyan/2342961.html&urlid=0" \t "http://www.shangxueba.com/jingyan/_blank)中进行查询操作，而且可以在文本文件、XML文件以及使用同一语法的数据源中查询数据。

## Fulltxt全文索引 与摘要索引

## 微服务soa化

独立分离的索引模块，可对数据库之外的数据建立索引

独立的sql解析查询模块，可对其他文件数据，内存集合对象数据，网络数据进行查询，类似linq

独立的存储引擎计划，api可参考mysql的存储引擎计划

## Other

### 自定义类型与类型值读取器

### 导入导出

### 可视化与报表

### Uml模型

### Web查询与更新编辑支持（这个好像已经sql2008有了，只需要实现了）

### Pagging翻页

### INFORMATION\_SCHEMA文件夹模式物理存储

### 驱动api标准化

数据库本地驱动api，参考mysql的。

实现jdbc odbc驱动接口

### ftp smb nfs支持，方便用户远程编辑记录

### ETL工具

### AS SERVICE

### Logic store逻辑as数据

# 参考

SQL的历史起源 - 张洪举 - 博客频道 - CSDN.NET.htm

SQL标准简介 - SQL Server(mssql)数据库栏目 - 红黑联盟.htm

SQL-92 - Wikipedia, the free encyclopedia.htm

SQL 1999 - Wikipedia, the free encyclopedia.htm

SQL 2003 - Wikipedia, the free encyclopedia.htm

SQL 2006 - Wikipedia, the free encyclopedia.htm

SQL 2008 - Wikipedia, the free encyclopedia.htm

SQL 2011 - Wikipedia, the free encyclopedia.htm

主流数据库技术的发展及未来方向

数据库前景广阔 五大趋势明显

LINQ：展望未来数据库发展趋势\_SQL Server - 百科教程网\_经验分享平台[上学吧经验教程频道].html