T-SQL

目次

目次

[１. RDBMS 4](#_Toc108285801)

[２. 入门篇 5](#_Toc108285802)

[２.１ 数据库组成 5](#_Toc108285803)

[２.１.１ 文件 5](#_Toc108285804)

[２.１.２ 文件组 5](#_Toc108285805)

[２.１.３ 系统数据库 5](#_Toc108285806)

[２.２ 数据库操作符 5](#_Toc108285807)

[２.２.１ 运算符优先级 5](#_Toc108285808)

[２.２.２ 算术运算符 6](#_Toc108285809)

[２.２.３ 比较运算符 6](#_Toc108285810)

[２.２.４ 逻辑运算符 6](#_Toc108285811)

[２.２.５ 位运算符 7](#_Toc108285812)

[２.２.６ 转义字符 7](#_Toc108285813)

[２.２.７ 集合运算符 7](#_Toc108285814)

[２.２.８ 转义符 7](#_Toc108285815)

[２.２.９ 数据库 8](#_Toc108285816)

[２.３ 数据类型 8](#_Toc108285817)

[２.４ 常量与变量 8](#_Toc108285818)

[２.４.１ 全局变量 8](#_Toc108285819)

[２.４.２ 局部变量 8](#_Toc108285820)

[２.４.３ 常量 8](#_Toc108285821)

[２.５ 语句优先级 8](#_Toc108285822)

[３. 初级篇 9](#_Toc108285823)

[３.１ 表操作 9](#_Toc108285824)

[３.１.１ DQL(数据查询语言) 9](#_Toc108285825)

[３.１.２ DML(数据操纵语言) 10](#_Toc108285826)

[３.１.３ DDL(数据定义语言) 10](#_Toc108285827)

[３.１.４ DCL(数据控制语言) 11](#_Toc108285828)

[３.１.５ 约束 11](#_Toc108285829)

[３.１.６ 临时表 12](#_Toc108285830)

[３.１.７ 子查询 12](#_Toc108285831)

[３.２ 迭代器 12](#_Toc108285832)

[３.３ 控制流 12](#_Toc108285833)

[３.４ 正则表达式 12](#_Toc108285834)

[３.４.１ 元字符 12](#_Toc108285835)

[４. 中级篇 13](#_Toc108285836)

[４.１ 索引 13](#_Toc108285837)

[４.１.１ 聚集索引 13](#_Toc108285838)

[４.１.２ 非聚集索引 13](#_Toc108285839)

[４.１.３ 唯一索引 13](#_Toc108285840)

[４.２ 约束 13](#_Toc108285841)

[４.３ 视图 13](#_Toc108285842)

[４.４ 游标 13](#_Toc108285843)

[４.４.１ 静态游标 13](#_Toc108285844)

[４.４.２ 动态游标 13](#_Toc108285845)

[４.４.３ 只进游标 13](#_Toc108285846)

[４.４.４ 键值驱动游标 13](#_Toc108285847)

[５. 高级篇 14](#_Toc108285848)

[５.１ 事务 14](#_Toc108285849)

[５.２ 锁 14](#_Toc108285850)

[５.３ 存储过程 14](#_Toc108285851)

[５.４ 触发器 14](#_Toc108285852)

# RDBMS

https://docs.microsoft.com/zh-cn/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms

|  |  |
| --- | --- |
| 数据库（Database） | 数据库是一些关联表的集合。 |
| 数据表（Table） | 表是数据的矩阵。表包含数据库中所有数据的数据库对象，有行和列组成。用于组织和存储数据。 |
| 记录（Record） | 表中的每一行称为一个记录，它由若个字段组成。 |
| 字段（Filed） | 表中的每一列称为一个字段，字段具有自己的属性，如字段大小，字段类型，字段类型决定了字段能存储那种数据。 |
| 索引（Record） | 使用索引可快速访问数据库表中的特定信息。索引是对数据库表中一列或多列的值进行排序的一种结构。类似于书籍的目录。 |
| 视图 | 视图是从一张表或多张表中导出表的(虚拟表)，是用户查看数据的一种方式，表中包括几个被定义的数据列和数据行。 |
| 存储过程 | 存储过程是一组为了完成特定功能的SQL语句集合（查询，插入，删除和更新等操作），经编译后以名称的形式存储在SQL Server服务器端的数据库中，有用户通过指定存储过程的名字来执行，当这个存储过程被调用执行时，这些操作也会同时执行。 |
| 主键 | 一列（或几列），其值能够唯一标识表中每一行。 |

# 入门篇

## 数据库组成

### 文件

|  |  |
| --- | --- |
| .mdf主要数据文件 | 存放数据和数据库的初始化信息，每个数据库有一个主要数据文件。 |
| .ndf次要数据文件 | 存放除主要数据文件以外的所有数据文件。有些数据库可能没有次要文件，也有可能有多个次要数据文件。 |
| .ldf | 此日志包含用于恢复数据库的信息。 每个数据库必须至少有一个日志文件。 |
| .bak | 数据库备份文件 |
| .iso | 镜像文件 |
| .trc |  |
| .xel |  |
| .udl | 数据链接UDL 文件存储 OLE DB 连接信息，例如提供程序、用户名、密码和其他选项 |
| .rpt | 包含由各种应用程序生成的报告 |
| .dsn | 数据源文件，存储数据库名称、目录、数据库驱动程序、用户 Id、密码等等连接详细信息 |
| .trn | 文件用于数据库还原，并可与其他.TRN 文件一起運用，以将回滚链接就任何從前的数据库状态。.TRN 文件凡是運用.BAK 文件进行备份，该文件是数据库的备份 |

### 文件组

### 系统数据库

|  |  |
| --- | --- |
| Master | 记录sql server实例的所有系统级信息，包括实例范围的元数据和端点，链接服务和系统配置设置。 |
| Tempdb | 临时数据库，用于保存临时对象或中间结果集。 |
| Model | 用作sql server实例上创建的所有数据库模版，对model数据库进行的修改（数据库大小，排序规则，恢复模式和其他数据库选项）将应用与以后创建的所有数据库。 |
| msdb | 用于sql server代理计划警报和作业。 |

## 数据库操作符

### 运算符优先级

|  |  |
| --- | --- |
| 1 | ~（位非）()(圆括号) |
| 2 | \*（乘）、/（除）、%（取模） |
| 3 | + (正值) ，- (负值) ，+ (加法) ，+ (串联) ，- (减法) ， & (位与) ，^ (位异或) ，| (位或) |
| 4 | =、 > 、 < 、 > =、 < =、 <> 、！ =、！ > 、！ < (比较运算符) |
| 5 | NOT |
| 6 | AND |
| 7 | ALL、ANY、BETWEEN、IN、LIKE、OR、SOME |
| 8 | =（赋值） |

### 算术运算符

|  |  |
| --- | --- |
| + | 加法 |
| - | 减法 |
| \* | 乘法 |
| /或DIV | 除法 |
| %或MOD | 取余 |

### 比较运算符

|  |  |
| --- | --- |
| = | 等于 |
| <>,!= | 不等于 |
| > | 大于 |
| < | 小于 |
| <= | 小于等于 |
| >= | 大于等于 |
| BETWEEN | 在指定的两个值之间 |
| Not BETWEEN | 不在指定范围 |
| IN | 在集合中 |
| NOT IN | 不在集合中 |
| <=> | 可以判断NULL和普通值是否相等 |
| LIKE | 模糊匹配 |
| Not like | 不在模糊匹配中 |
| REGEXP或RLIKE | 正则式匹配 |
| Is Null | 为空 |
| is not null | 不为空 |
| !< | 不小于 |
| !> | 不大于 |
| != | 不等于 |

### 逻辑运算符

|  |  |
| --- | --- |
| ALL | 如果一组的比较都为 TRUE，那么就为 TRUE。 |
| AND | 如果两个布尔表达式都为 TRUE，那么就为 TRUE。 |
| ANY | 如果一组的比较中任何一个为TRUE，那么就为 TRUE。 |
| BETWEEN | 如果操作数在某个范围之内，那么就为 TRUE。 |
| EXISTS | 指定一个子查询，测试行是否存在。 |
| IN | 如果操作数等于表达式列表中的一个，那么就为 TRUE。 |
| LIKE | 如果操作数与一种模式相匹配，那么就为 TRUE。 |
| NOT | 对任何其他布尔运算符的值取反。 |
| OR | 如果两个布尔表达式中的一个为 TRUE，那么就为 TRUE。 |
| SOME | 如果在一组比较中，有些为 TRUE，那么就为 TRUE。 |

### 符合运算符

|  |  |
| --- | --- |
| +=（加法赋值） | 将原始值加上一定的量，并将原始值设置为结果。 |
| -=（减法赋值） | 将原始值减去一定的量，并将原始值设置为结果。 |
| \*=（乘法赋值） | 将原始值乘上一定的量，并将原始值设置为结果。 |
| /=（除法赋值） | 将原始值除以一定的量，并将原始值设置为结果。 |
| %=（取模赋值） | 将原始值除以一定的量，并将原始值设置为余数。 |
| &=（位与赋值） | 对原始值执行位与运算，并将原始值设置为结果。 |
| ^=（位异或赋值） | 对原始值执行位异或运算，并将原始值设置为结果。 |
| |=（位异或赋值） | 对原始值执行位或运算，并将原始值设置为结果。 |

### 位运算符

|  |  |
| --- | --- |
| &（位与） | &=（位与赋值） |
| |（位或） | |=（位或赋值） |
| ^（位异或） | ^=（位异或赋值） |
| ~（位非） |  |

### 转义字符

|  |  |
| --- | --- |
| \_下划线 | 表示匹配任意一个字符 |
| \*通配符 | 表示匹配任意字符 |
| %百分号 | 表示匹配零个及多个任意字符 |
| \取消转义 | \表示特殊符号不进行转义，当做普通字符 |
| []匹配范围 | 匹配一个范围 |
| [^]排除一个范围 | 排除一个范围 |
| / |  |

### 集合运算符

|  |  |
| --- | --- |
| EXCEPT | 从左侧输入查询返回非由右侧输入查询输出的非重复行。 |
| INTERSECT | 返回由左右双侧输入查询运算符输出的非重复行。 |
| UNION ALL | 连接两个查询中的结果集。 包括重复行 |
| UNION | 连接两个查询中的结果集。 排除重复行 |

### 转义符

|  |  |
| --- | --- |
| []分隔标识符 | 当转义符置于通配符时，关键字解释为普通字符 |

## 数据类型

## 常量与变量

### 全局变量

定义：以@@开头，后面跟相应的字符串

使用全局变量时应该注意以下几点：

①全局变量不是由用户的程序定义的，它们是在服务器级定义的。

②用户只能使用预先定义的全局变量。

③引用全局变量时，必须以标记符“@@”开头。

④局部变量的名称不能与全局变量的名称相同，否则会在应用程序中出现不可预测的结果。

### 局部变量

局部变量是以@开头，后面是变量名。作用域范围仅在程序内部。关键字是：declare

### 常量

## 语句优先级

|  |  |
| --- | --- |
| FROM | 执行顺序为从后往前、从右到左。数据量较大的表尽量放在后面。 |
| WHERE | 执行顺序为自下而上、从右到左。将能过滤掉最大数量记录的条件写在WHERE字句的最右。 |
| GROUP BY | 执行顺序从右往左分组，最好在GROUP BY前使用WHERE将不需要的记录在GROUP BY之前过滤掉 |
| HAVING | 消耗资源。尽量避免使用，HAVING会在检索出所有记录之后才对结果进行过滤，需要排序等操作。 |
| SELECT | 少用\*号，尽量使用字段名称，oracle在解析的过程中，通过查询数据字典将\*号依次转换成所有列名，消耗时间。 |
| ORDER BY | 执行顺序从左到右，消耗资源 |

## 表操作

### DQL(数据查询语言)

数据查询语言DQL基本结构是由SELECT子句，FROM子句，WHERE子句组成的查询块：

SELECT <字段名表>

FROM <表或视图名>

WHERE <查询条件>

#### 内连接(INNER JOIN)

内部连接是指根据连接条件对两个或多个表的连接字段进行比较，将符合连接条件的记录连接起来的一种连接形式。返回两张表都匹配的行，不匹配的行不显示。

#### 等值连接

使用等号“=”进行的连接称为等值连接”

#### 非等值连接

使用等号“=”以外的比较运算符，如“<、>、<>”进行的连接称为“非等值连接”

#### 自连接

自连接是一张表与自己连接，自连接只支持inner join和left join

#### 外连接

外连接是将符合条件的行进行连接，而外连接将会保留某个表或者视图的所有行以及另外一个表中不符合条件的行。

##### 左外部连接（LEFT OUTER JOIN）

左外连接保留左边表的所有记录以及右边表中符合条件的记录。如果左表的某一记录在右表中没有匹配行，右表中相应的列显示为null。

##### 右外部连接（RIGHT OUTER JOIN）

右外连接保留右边表的所有记录以及左边表中符合条件的记录。如果右表的某一记录在左表中没有匹配行，左表中相应的列显示为null。

##### 全外连接（FULL OUTER JOIN）

完全外连接保留左表和右表所有记录，如果左表某一记录在右表中无匹配行，右表相应列 就显示为 NULL 如果右表某 记录在左表中无匹配行，左表相应列就显示为 NULL。

##### 交叉连接（CROSS JOIN）

交叉连接结果为两个表的笛卡尔积，结果集中的行数为两个表行数的积，结果集中的列数 为两个表列的和。

##### 表的相关数据

|  |  |
| --- | --- |
| 字段 | 字段是一个事物（对象）的某一个静态特征（属性） |
| 记录 | 记录（对象）是字段的组合，表示一个具体的事物（对象） |
| 表 | 表是记录的集合，表示同一类事物（对象）的集合 |
| 列 | 列是字段的另一种称谓 |
| 属性 | 字段的另一种称谓 |
| 元祖 | 记录的另一种称谓 |
| 主键 | （唯一识别码）唯一标识表的属性（防止表冗余） |
| 外键 | 主表的主关键字为从表的外键 |
| 外键表 | 有外键的表称为外键表 |
| 主键表 | 与外键表具有相同字段的表，外键表的属性来源于主键表 |
| 约束 | 限制，保证数据合理 |

#### 联合操作符（UNION）

联合操作符是一种用于两个或多个查询产生一个结果集的特殊操作符UNION相当于是将两个结果集加起来。

将一个结果集添加到另一个结果集的末尾。JOIN可以认为是将表进行水平组合，而UNION则是将表进行垂直组合。

### DML(数据操纵语言)

|  |  |
| --- | --- |
| Select | 查询表数据 |
| INSERT INTO VALUES() | 向表插入数据 |
| Delete | 删除表数据 |
| update | 更新表数据 |

### DDL(数据定义语言)

数据定义语言DDL用来创建数据库中的各种对象-----表、视图、索引、同义词、聚簇等如：

CREATE TABLE/VIEW/INDEX/SYN/CLUSTER，进行创建，删除，修改

DDL操作是隐性提交的！不能rollback

|  |  |
| --- | --- |
|  |  |

### DCL(数据控制语言)

数据控制语言DCL用来授予或回收访问数据库的某种特权，并控制数据库操纵事务发生的时间及效果，对数据库实行监视等。如：

1) GRANT：授权。

2) ROLLBACK [WORK] TO [SAVEPOINT]：回退到某一点。

回滚---ROLLBACK

回滚命令使数据库状态回到上次最后提交的状态。其格式为：

SQL>ROLLBACK

|  |  |
| --- | --- |
| Grant | 授予用户访问权限 |
| Deny | 拒绝用户访问 |
| revoke | 删除用户访问权限 |

### 约束

|  |  |
| --- | --- |
| 非空约束(NOT NULL) |  |
|  |  |

#### 非空约束(NOT NULL)

**使用户必须在表的指定列中输入一个值。每个表中可以有多个非空约束。**

#### 检查约束(CHECK)

**用来指定一个布尔操作，限制输入到表中的值。**

#### 唯一约束(UNIQUE)

使用户的应用程序必须向列中输入一个唯一值，值不能重复，但可以为空。

#### 主键约束(PRIMARY KEY)

**建立一列或多列的组合以唯一标识表中的每一行。主键可以保证数据完整性，一个表只有一个主键，同时主键中的列不能接受空值。**

#### 外键约束(FOREIGN KEY)

**外键是用于建立和加强两个表数据之间的链接的一列或多列。当一个表中作为主键的一列被添加到另一个表中时，链接就建立了，主要目的时控制存储在外键表中的数据。**

### 临时表

### 子查询

子查询是一个嵌套在select，insert，Update，Delect语句或其他子查询中的查询，任何允许使用表达式的地方都可以使用子查询。

# 初级篇

## 迭代器

## 控制流

|  |  |
| --- | --- |
| Case when then when then else end |  |
| If else |  |
| BEGIN…END |  |
| WHILE |  |
| GOTO |  |
| Return |  |
| Waitfor |  |
| WHILE　continue break |  |

## 正则表达式

### 元字符

|  |  |
| --- | --- |
| ^ 文本的开始 | $ 文本的结尾 |
| [[:<:]] 词的开始 | [[:>:]] 词的结尾 |

## Metadata（元数据）

|  |  |
| --- | --- |
| OBJECT\_ID | 返回架构范围内对象的数据库对象标识号。 |
| SERVERPROPERTY | 返回有关服务器实例的属性信息 |

## 系统

|  |  |
| --- | --- |
| NEWID | 随机返回uniqueidentifier 类型的唯一值。 |
| ISNULL | 使用指定的替换值替换 NULL。 |

## 转换

|  |  |
| --- | --- |
|  |  |

## 表达式

|  |  |
| --- | --- |
| COALESCE | 按顺序计算变量并返回最初不等于 NULL 的第一个表达式的当前值 |
|  |  |

## 窗口函数

对于基本查询中的每一行，按行的窗口（组）进行运算，并计算一个标量（单个）结果值。开窗函数对由OVER子句公开的一组行进行运算。对于基本查询中的每一行，OVER子句公开给函数的是来自基本查询结果集的一个行子集。

# 中级篇

## 索引

### 聚集索引

### 非聚集索引

### 唯一索引

## 约束

### 主键约束

### 唯一约束

### 外键约束

### CHECK约束

### 默认约束

### 非空约束

使用户必须在表的指定列中输入一个值。每个表中可以有多个非空约束。

## 视图

## 游标

### 静态游标

### 动态游标

### 只进游标

### 键值驱动游标

## 事务

### 自动提交事务

每条单独的语句都是一个事务。

### 显式事务

每个事务均以 BEGIN TRANSACTION 语句显式开始，以 COMMIT 或 ROLLBACK 语句显式结束。

### 隐式事务

在前一个事务完成时新事务隐式启动，但每个事务仍以 COMMIT 或 ROLLBACK 语句显式完成。

### 批处理级事务

只能应用于多个活动结果集 (MARS)，在 MARS 会话中启动的 Transact-SQL 显式或隐式事务变为批处理级事务。 当批处理完成时没有提交或回滚的批处理级事务自动由 SQL Server 进行回滚。

## 锁

## 存储过程

## 触发器

# 高级篇

## 设计范式

### 第一设计范式

第一范式是说关系（表）中的元组（行）必须是唯一的，并且属性是原子化的。通过为表定义一个唯一键就可以实现唯一行。

### 第二设计范式

### 第三设计范式

### 第四设计范式

# SSMS

<https://docs.microsoft.com/zh-cn/sql/ssms/sql-server-management-studio-keyboard-shortcuts?view=sql-server-ver16>

快捷键

## 登录

登录ssms有以下几种方式；使用最多的是服务器引擎；服务器名可以使用本地电脑名作为服务器；(Local);使用[.]；在工作中一般把数据库架构在远程服务器或云端中，在这时服务器名药使用远程服务器名，服务器IP，服务器证明书登录，访问云端的公有IP；在虚拟环境中要使用公网ip相对应的私有IP，服务器名登录。

## 日志-Log

## SQL Server 配置管理器

SQL Server 2019 Configuration Manager

SQLServerManager15.msc

services.msc命令打开服务管理器停止sqlserver引擎。