**SQLSERVER数据库的T-SQL基本语法**

**T-SQL Basic Syntax**

# 01、T-SQL的定义

**T-SQL：** 结构化查询语言(Transact-Structured Query Language)，SQL在Microsoft SQL Server上的增强版；

**组成部分：** 数据定义语言(DDL)/数据操作语言(DML)/数据控制语言(DCL)/事务控制语言(TCL)/附加语言元素；

# 03、数据定义语言(DDL, Data Definition Language)

用于定义和管理SQL数据库中的所有对象的语言；

|  |  |  |
| --- | --- | --- |
| **语句** | **功能** | **示例** |
| **CREATE** | 创建； | CREATE DATABASE|TABLE|INDEX|PROCEDURE|FUNCTION|VIEW|TRIGGER; |
| **ALTER** | 修改； | … |
| **DROP** | 删除； | … |
| **TRUNCATE** | 清空； | -- 清空表中所有记录(DDL);  TRUNCATE TABLE StuInfo; |
| **COMMENT** | 注释； | EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'商品序号', @level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level1name=N'ProductBasic', @level2type=N'COLUMN', @level2name=N'ProductId' |

# 04、数据操作语言(DML, Data Manipulation Language)

SQL中处理数据等操作统称为数据操纵语言；

|  |  |  |
| --- | --- | --- |
| **语句** | **功能** | **示例** |
| **SELECT** | 查询； | -- 查询表中所有列;  SELECT \* FROM StuInfo;  -- 查询表中某些列的数据;  SELECT StuId, StuName, StuSex FROM StuInfo ;  -- 查询表中某一列的数据;  SELECT StuId FROM StuInfo; |
| **INSERT** | 插入； | -- 对应插入;  INSERT INTO StuInfo(StuId, StuName, StuSex) VALUES(1, 'XXX01', '男');  -- 全部插入;  INSERT INTO StuInfo VALUES(1, 'XXX02', '男');  -- SELECT INTO;  SELECT \* INTO StuInfo2  FROM StuInfo  WHERE StuSex = '女'  -- INSERT INTO SELECT;  INSERT INTO StuInfo2(StuName, StuSex)  SELECT StuName, StuSex FROM StuInfo; |
| **UPDATE** | 更新； | -- 单个字段更新;  UPDATE StuInfo SET StuName = 'XXX03' WHERE StuId = 1;  -- 多个字段更新;  UPDATE StuInfo SET StuName = 'XXX03',StuSex = '女' WHERE StuId = 1;  -- 给某个字段加5更新;  UPDATE StuInfo SET Score = Score + 5 WHERE StuId = 1; |
| **DELETE** | 删除； | -- 删除一行记录;  DELETE FROM StuInfo WHERE StuId = 1; |
| **CALL** |  | … |
| **EXPLAIN PLAN** |  | … |
| **LOCK TABLE** |  | … |

# 05、数据控制语言(DCL, Data Control Language)

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，在默认状态下，只有sysadmin,dbcreator,db\_owner或db\_securityadmin等人员才有权力执行DCL命令；

|  |  |  |
| --- | --- | --- |
| **语句** | **功能** | **示例** |
| **GRANT** | 授权语句，把权限授予给其他用户和角色； | … |
| **DENY** | 拒绝给当前数据库内的用户或者角色授予权限，并防止用户或角色通过其组或角色成员继承权限； | … |
| **REVOKE** | 将以前在当前数据库内的用户或者角色上授予或拒绝的权限删除，但是该语句并不影响用户或者角色从其他角色中作为成员继承过来的权限； | … |

# 06、事务控制语言(TCL, Transaction Control Language)

当一个事务执行并完成修改时，并不是对目标表立即进行修改，此时修改的结果只是保存到临时缓存中，只有利用事务控制命令才最终认可这个事务；

|  |  |  |
| --- | --- | --- |
| **语句** | **功能** | **示例** |
| **COMMIT** | 用于把事务所做的修改保存到数据库，表面该事务对数据库所做的操作将永久记录到数据库；； | BEGIN TRAN  -- 已更新;  UPDATE scott\_emp SET sal = sal \* 2;  -- 提交事务;  COMMIT TRAN |
| **ROLLBACK** | 用于撤销还没有被保存到(未提交的事务)数据库的命令； | BEGIN TRAN  -- 已更新;  UPDATE scott\_emp SET sal = sal \* 2;  -- 撤销事务;  ROLLBACK TRAN |
| **SAVEPOINT** | 保存点是事务过程中的一个逻辑点，可以把事务回退到这个点，而不必回退整个事务； | BEGIN TRAN  INSERT INTO scott\_emp VALUES(100.00)  SAVE TRAN point1  INSERT INTO scott\_emp VALUES(200.00)  ROLLBACK TRAN point1  COMMIT TRAN |
| **SET TRANSACTION** | 设置事务隔离级别； | SET TRANSACTION ISOLATION LEVEL READ COMMITTED  BEGIN TRAN  UPDATE scott\_emp SET sal = sal \* 2;  COMMIT TRAN |

# 07、附加语言元素

## 05.01、变量

**A、局部变量：@**

-- 声明局部变量;

DECLARE @变量名1 变量类型1, @变量名2 变量类型2...

-- 局部变量赋值;

SELECT @变量名 = 值

SET @变量名 = 值

-- 使用聚合函数语为变量赋值;

DECLARE @grade INT

SELECT @grade = MAX(Grade) FROM Score

PRINT @grade

--在查询语句中为变量赋值;

DECLARE @EnglishID NVARCHAR(50), @StoreTitle NVARCHAR(50)

SELECT

@EnglishID = EnglishID, @StoreTitle = StoreTitle

FROM

StoreBasic

PRINT @EnglishID + '-' + @StoreTitle

-- test036-test036

**B、全局变量：@@**

SQL Server系统内部使用的变量，存储SQL Server的配置设定值和效率统计数据等；

@@ERROR： 返回上一个执行的T-SQL语句的错误号，若没有错误，那么返回0；

@@SERVERNAME： 返回运行SQL Server的本地服务器的名称；

@@ROWCOUNT： 返回受上一语句影响的行数；

......

## 05.02、运算符

运算符用来执行列、常量或变量之间的数学运算和比较操作；

|  |  |  |
| --- | --- | --- |
| **类型** | **运算符** | **说明** |
| **算术运算符** | **+** | 加法 |
| **-** | 减法 |
| **\*** | 乘法 |
| **/** | 除法 |
| **%** | 取模，两个整数相除后的余数； |
| **位运算符** | **&** | 按位与，两个位均=1时，结果=1，否则=0； |
| **|** | 按位或，只要一个位为1，结果=1，否则=0； |
| **^** | 按位异或，两个位值不同时，结果=1，否则=0； |
| **比较运算符** | **=** | 相等 |
| **>** | 大于 |
| **<** | 小于 |
| **>=** | 大于或等于 |
| **<=** | 小于或等于 |
| **<>** | 不等于 |
| **!=** | 不等于 |
| **!<** | 不小于 |
| **!>** | 不大于 |
| **逻辑运算符** | **AND** | 如果两个操作数值都=TRUE，结果=TRUE； |
| **OR** | 如果两个操作数值有一个=TRUE，结果=TRUE； |
| **NOT** | 如果一个操作数值=TRUE，结果=FALSE，否则=TRUE； |
| **ALL** | 如果每个操作数值=TRUE，结果=TRUE； |
| **ANY** | 在一系列操作数值中只要有一个=TRUE，结果=TRUE； |
| **BETWEEN** | 如果操作数值在指定的范围内，结果=TRUE； |
| **EXISTS** | 如果子查询包含一些行，结果=TRUE； |
| **IN** | 如果操作数值等于表达式列表中的一个，结果=TRUE； |
| **LIKE** | 如果操作数值与一种模式相匹配，结果=TRUE； |
| **SOME** | 如果在一些操作数值中，有些数值=TRUE，结果=TRUE； |
| **字符串连接运算符** | **+** | 形式上与算术运算符"+"是完全一致的，但用于字符串时表示字符串连接； |
| **赋值运算符** | **=** | 形式上与算术运算符"="是完全一致的，但用于变量时表示赋值； |
| **优先级** | **运算符** | **说明** |
| **1** | **()** | 小括号； |
| **2** | **+、-、~** | 正、负、逻辑非； |
| **3** | **\*、/、%** | 乘法、除法、取模； |
| **4** | **+、-、+** | 加法、减法、字符串连接； |
| **5** | **比较运算符** | 各个比较运算符； |
| **6** | **位运算符** | 各个位运算符； |
| **7** | **NOT** | 逻辑非； |
| **8** | **AND** | 逻辑与； |
| **9** | **其他逻辑运算符** | 其他逻辑运算符； |
| **10** | **赋值运算符** | 赋值运算符; |

## 05.03、系统函数(System Functions)

### 05.03.01、聚合函数(Aggregate Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **AVG(column)** | 返回某列的平均值； | -- 语法;  SELECT AVG(column\_name) FROM table\_name;  -- 某字段的平均值;  SELECT AVG(OrderPrice) AS OrderAverage FROM Orders; |
| **BINARY\_CHECKSUM()** |  |  |
| **CHECKSUM()** |  |  |
| **CHECKSUM\_AGG()** |  |  |
| **COUNT(…)** | 返回行数； | -- 返回指定列的值的数目(NULL不计);  SELECT COUNT(column\_name) FROM table\_name;  -- 返回表中的记录数;  SELECT COUNT(\*) FROM table\_name;  -- 返回指定列的不同值的数目;  SELECT COUNT(DISTINCT column\_name) FROM table\_name;  -- 返回指定列的值的数目(NULL不计);  SELECT  COUNT(Customer) AS CustomerNilsen  FROM Orders  WHERE Customer = 'Carter';  -- 返回表中的记录数;  SELECT COUNT(\*) AS Numbers  FROM Orders;  -- 返回指定列的不同值的数目;  SELECT COUNT(DISTINCT Customer) AS Numbers  FROM Orders; |
| **COUNT\_BIG()** |  |  |
| **GROUPING()** |  |  |
| **GROUPING\_ID()** |  |  |
| **FIRST(column)** | 返回某列的第一个值； | -- 语法;  SELECT FIRST(column\_name) FROM table\_name;  -- 某列的第一个值;  SELECT FIRST(OrderPrice) AS FirstOrderPrice FROM Orders; |
| **LAST(column)** | 返回某列的最后一个值； | -- 语法;  SELECT LAST(column\_name) FROM table\_name;  -- 某列的最后一个值;  SELECT LAST(OrderPrice) AS LastOrderPrice FROM Orders; |
| **MAX(column)** | 返回某列的最大值； | -- 语法;  SELECT MAX(column\_name) FROM table\_name;  -- 某列的最大值;  SELECT MAX(OrderPrice) AS LargestOrderPrice FROM Orders; |
| **MIN(column)** | 返回某列的最小值； | -- 语法;  SELECT MIN(column\_name) FROM table\_name;  -- 某列的最小值;  SELECT MIN(OrderPrice) AS SmallestOrderPrice FROM Orders; |
| **STDEV()** |  |  |
| **STDEVP()** |  |  |
| **SUM(column)** | 返回某列的总和； | -- 语法;  SELECT SUM(column\_name) FROM table\_name;  -- 某列的总和;  SELECT SUM(OrderPrice) AS OrderTotal FROM Orders; |
| **VAR()** |  |  |
| **VARP()** |  |  |

### 05.03.02、配置函数(Configuration Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **CONNECTIONPROPERTY()** |  |  |
| **@@DATEFIRST** |  |  |
| **@@DBTS** |  |  |
| **@@LANGID** |  |  |
| **@@LANGUAGE** |  |  |
| **@@LOCK\_TIMEOUT** |  |  |
| **@@MAX\_CONNECTIONS** |  |  |
| **@@MAX\_PRECISION** |  |  |
| **@@NESTLEVEL** |  |  |
| **@@OPTIONS** |  |  |
| **@@REMSERVER** |  |  |
| **@@SERVICENAME** |  |  |
| **@@SPID** |  |  |
| **@@TEXTSIZE** |  |  |
| **@@VERSION** |  |  |

### 05.03.03、游标函数(Cursor Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **@@CURSOR\_ROWS** |  |  |
| **CURSOR\_STATUS()** |  |  |
| **@@FETCH\_STATUS** |  |  |

### 05.03.04、日期和时间函数(Date And Time Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **GETDATE()** | 返回当前日期和时间； | -- 语法;  GETDATE()  -- 获取当前日期;  SELECT GETDATE() AS CurrentDateTime;  -- 字段设为默认值;  CREATE TABLE Orders  (  OrderId int NOT NULL PRIMARY KEY,  ProductName varchar(50) NOT NULL,  OrderDate datetime NOT NULL DEFAULT GETDATE()  ); |
| **GETUTCDATE()** | 返回当前UTC时间(世界时间坐标或格林尼治标准时间)； | -- 语法;  GETUTCDATE();  -- 获取当前日期;  SELECT GETUTCDATE() AS CurrentDateTime; |
| **DATEPART()** | 返回日期/时间的单独部分； | -- 语法;  DATEADD(datepart, number, date);  -- 获取当前日期的年月日;  SELECT  DATEPART(yyyy, GETDATE()) AS OrderYear,  DATEPART(mm, GETDATE()) AS OrderMonth,  DATEPART(dd, GETDATE()) AS OrderDay  -- 某个字段的年月日;  SELECT  DATEPART(yyyy, OrderDate) AS OrderYear,  DATEPART(mm, OrderDate) AS OrderMonth,  DATEPART(dd, OrderDate) AS OrderDay  FROM Orders  WHERE OrderId = 1 |
| **DATEADD()** | 在日期中添加/减去指定的时间间隔； | -- 语法;  DATEADD(datepart, number, date);  -- 获取当前日期的后2天;  SELECT DATEADD(dd, 2, GETDATE());  -- 某个字段的后2天;  SELECT  OrderId, DATEADD(dd, 2, OrderDate) AS OrderPayDate  FROM Orders |
| **DATEDIFF()** | 返回两个日期之间的时间； | -- 语法;  DATEDIFF(datepart, startdate, enddate);  -- 两个日期的天数;  SELECT DATEDIFF(dd, '2008-12-29', '2008-12-30') AS DiffDate; |
| **CONVERT()** | 用不同的格式显示日期/时间； | -- 语法;  CONVERT(data\_type(length), data\_to\_be\_converted, style);  -- 获取当前日期;  SELECT CONVERT(VARCHAR(19), GETDATE());  SELECT CONVERT(VARCHAR(10), GETDATE(), 110);  SELECT CONVERT(VARCHAR(11), GETDATE(), 106);  SELECT CONVERT(VARCHAR(24), GETDATE(), 113); |
| **YEAR()** | 返回指定日期中的年份的整数； | -- 语法;  YEAR(date);  -- 获取当前日期的年;  SELECT YEAR(GETDATE()); |
| **MONTH()** | 返回指定日期中的月份的整数； | -- 语法;  MONTH(date);  -- 获取当前日期的月;  SELECT MONTH(GETDATE()); |
| **DAY()** | 返回指定日期中的天的整数； | -- 语法;  DAY(date);  -- 获取当前日期的天;  SELECT DAY(GETDATE()); |
| **Datepart枚举** | **格式** | |
| 年 | yy, yyyy | |
| 季度 | qq, q | |
| 月 | mm, m | |
| 年中的日 | dy, y | |
| 日 | dd, d | |
| 周 | wk, ww | |
| 星期 | dw, w | |
| 小时 | hh | |
| 分钟 | mi, n | |
| 秒 | ss, s | |
| 毫秒 | ms | |
| 微妙 | mcs | |
| 纳秒 | ns | |
| **Style枚举** | **格式** | |
| 100或者0 | mon dd yyyy hh:miAM(PM） | |
| 101 | mm/dd/yy | |
| 102 | yy.mm.dd | |
| 103 | dd/mm/yy | |
| 104 | dd.mm.yy | |
| 105 | dd-mm-yy | |
| 106 | dd mon yy | |
| 107 | Mon dd, yy | |
| 108 | hh:mm:ss | |
| 109或者9 | mon dd yyyy hh:mi:ss:mmmAM(PM) | |
| 110 | mm-dd-yy | |
| 111 | yy/mm/dd | |
| 112 | yymmdd | |
| 113或者13 | dd mon yyyy hh:mm:ss:mmm(24h) | |
| 114 | hh:mi:ss:mmm(24h) | |
| 120或者20 | yyyy-mm-dd hh:mi:ss(24h) | |
| 121或者21 | yyyy-mm-dd hh:mi:ss.mmm(24h) | |
| 126 | yyyy-mm-ddThh:mm:ss.mmm | |
| 130 | dd mon yyyy hh:mi:ss:mmmAM | |
| 131 | dd/mm/yy hh:mi:ss:mmmAM | |

### 05.03.05、数学函数(Mathematical Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **ABS(x)** | 返回绝对值； | SELECT ABS(2), ABS(-3.3), ABS(-33);  -- 2, 3.3, 33 |
| **ACOS(x)** | (反余弦)返回x的反余弦,即余弦x的值，若x不在-1到1范围之内，则返回NULL； | SELECT ACOS(1), ACOS(0);  -- 0, 1.5707963267949 |
| **ASIN(x)** | (反正弦)返回x的反正弦，即正弦x的值，若x不在-1到1的范围内，则返回NULL； | SELECT ASIN(1), ASIN(0);  -- 1.5707963267949, 0 |
| **ATAN(x)** | (反正切)返回x的反正切，即正切x的值； | SELECT ATAN(0.30933624960962325), ATAN(1);  -- 0.3, 0.785398163397448 |
| **ATN2()** |  |  |
| **CEILING(x)** | (取整)返回不小于x的最小整数值； | SELECT CEILING(-3.35), CEILING(3.35);  -- -3, 4 |
| **COS(x)** | (余弦)返回x的余弦，其中x为弧度值； | SELECT COS(0), COS(PI());  -- 1, -1 |
| **COT(x)** | (余切)返回x的余切； | SELECT COT(0.3), 1/TAN(0.3), COT(PI()/4);  -- 3.23272814376583, 3.23272814376583, 1 |
| **DEGREES(x)** | (角度与弧度相互转换)参数x有弧度转换为角度； | SELECT DEGREES(PI()/2), DEGREES(PI());  -- 90, 180 |
| **EXP(x)** | (幂运算)返回e的x乘方后的值； | SELECT EXP(3), EXP(-3), EXP(0);  -- 20.0855369231877, 0.0497870683678639, 1 |
| **FLOOR(x)** | (取整)返回不小于x的最大整数值； | SELECT FLOOR(-3.35), FLOOR(3.35);  -- -4, 3 |
| **LOG(x)** | (对数运算)返回x的自然对数，x相对于基数e的对数； | SELECT LOG(3), LOG(6);  -- 1.09861228866811, 1.79175946922805 |
| **LOG10(x)** | (对数运算)返回x的基数为10的对数； | SELECT LOG10(1), LOG10(100), LOG10(1000);  -- 0, 2, 3 |
| **PI()** | 返回圆周率的值； | SELECT PI();  -- 3.14159265358979 |
| **POWER(x, y)** | (幂运算)返回x的y次乘方的结果值； | SELECT POWER(2, 2), POWER(2.00, -2);  -- 4, 0.25 |
| **RANDIANS(x)** | (角度与弧度相互转换)参数x由角度转换为弧度； | SELECT RADIANS(90.0), RADIANS(180.0);  -- 1.570796326794896600, 3.141592653589793100 |
| **RAND(…)** | 返回随机数； | -- 不指定种子值;  SELECT RAND(), RAND(), RAND();  -- 0.928038620073639, 0.213602969191392, 0.057220753910619  -- 指定种子值;  SELECT RAND(10), RAND(10), RAND(11);  -- 0.713759689954247, 0.713759689954247, 0.713778322925506 |
| **ROUND(x, y)** | 四舍五入； | SELECT ROUND(1.38, 1), ROUND(1.38, 0), ROUND(232.38, -1), ROUND(232.38, -2);  -- 1.40, 1.00, 230.00, 200.00 |
| **SIGN(x)** | 返回符号； | SELECT SIGN(-21), SIGN(0), SIGN(21);  -- -1, 0, 1 |
| **SIN(x)** | (正弦)返回x的正弦，其中x为弧度值； | SELECT SIN(PI()/2), SIN(PI());  -- 1, 1.22464679914735E-16 |
| **SQRT(x)** | 返回平方根； | SELECT SQRT(9), SQRT(40);  -- 3, 6.32455532033676 |
| **SQUARE(x)** | (幂运算)返回指定浮点值x的平方； | SELECT SQUARE(3), SQUARE(-3), SQUARE(0);  -- 9, 9, 0 |
| **TAN(x)** | (正切)返回x的正切，其中x为给定的弧度值； | SELECT TAN(0.3),ROUND(TAN(PI()/2),0);  -- 0.309336249609623, 1.63312393531954E+16 |

### 05.03.06、元数据函数(Metadata Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **COL\_LENGTH()** |  |  |
| **COL\_NAME()** |  |  |
| **COLUMNPROPERTY()** |  |  |
| **DATABASEPROPERTY()** |  |  |
| **DATABASEPROPERTYEX()** |  |  |
| **DB\_ID()** |  |  |
| **DB\_NAME()** |  |  |
| **FILE\_ID()** |  |  |
| **FILE\_NAME()** |  |  |
| **FILEGROUP\_ID()** |  |  |
| **FILEGROUP\_NAME()** |  |  |
| **FILEGROUPPROPERTY()** |  |  |
| **FILEPROPERTY()** |  |  |
| **::FN\_LISTEXTENDEDPROPERTY()** |  |  |
| **FULLTEXTCATALOGPROPERTY()** |  |  |
| **FULLTEXTSERVICEPROPERTY()** |  |  |
| **INDEX\_COL()** |  |  |
| **INDEXKEY\_PROPERTY()** |  |  |
| **INDEXPROPERTY()** |  |  |
| **OBJECT\_ID()** |  |  |
| **OBJECT\_NAME()** |  |  |
| **OBJECTPROPERTY()** |  |  |
| **OBJECTPROPERTYEX()** |  |  |
| **@@PROCID** |  |  |
| **SQL\_VARIANT\_PROPERTY()** |  |  |
| **TYPEPROPERTY()** |  |  |
| **CHANGE\_TRACKING\_CURRENT\_VERSION()** |  |  |
| **CHANGE\_TRACKING\_IS\_COLUMN\_IN\_MASK()** |  |  |
| **CHANGE\_TRACKING\_CLEANUP\_VERSION()** |  |  |

### 05.03.07、其他函数(Other Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **APP\_NAME()** |  |  |
| **CAST()** |  |  |
| **COALESCE()** |  |  |
| **COLLATIONPROPERTY()** |  |  |
| **CONVERT()** |  |  |
| **CURRENT\_USER** |  |  |
| **DATALENGTH()** |  |  |
| **@@ERROR** |  |  |
| **FN\_HELPCOLLATIONS()** |  |  |
| **FN\_INDEXINFO()** |  |  |
| **::FN\_SERVERSHAREDDRIVERS()** |  |  |
| **::FN\_VIRTUALSERVERNODES()** |  |  |
| **FORMATMESSAGE()** |  |  |
| **GETANSINULL()** |  |  |
| **HOST\_ID()** |  |  |
| **HOST\_NAME()** |  |  |
| **IDENT\_CURRENT()** |  |  |
| **IDENT\_INCR()** |  |  |
| **IDENT\_SEED()** |  |  |
| **@@IDENTITY** |  |  |
| **IDENTITY()(SELECT INTO)** |  |  |
| **ISNULL()** |  |  |
| **ISNUMERIC()** |  |  |
| **NEWID()** |  |  |
| **NULLIF()** |  |  |
| **PARSENAME()** |  |  |
| **PERMISSIONS()** |  |  |
| **@@ROWCOUNT** |  |  |
| **ROWCOUNT\_BIG()** |  |  |
| **SCOPE\_IDENTITY()** |  |  |
| **SERVERPROPERTY()** |  |  |
| **SESSIONPROPERTY()** |  |  |
| **SESSION\_USER** |  |  |
| **STATS\_DATE()** |  |  |
| **SYSTEM\_USER** |  |  |
| **@@TRANCOUNT** |  |  |
| **UPDATE()** |  |  |
| **USER\_NAME()** |  |  |

### 05.03.08、层次结构ID函数(Hierarchy Id Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **GETROOT()** |  |  |
| **GETDESCENDANT()** |  |  |
| **ISDESCENDANTOF()** |  |  |
| **GETLEVEL()** |  |  |
| **GETANCESTOR()** |  |  |
| **GETREPARENTEDVALUE()** |  |  |
| **TOSTRING()** |  |  |

### 05.03.09、行集函数(Rowset Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **CONTAINSTABLE()** |  |  |
| **FREETEXTTABLE()** |  |  |
| **OPENDATASOURCE()** |  |  |
| **OPENQUERY()** |  |  |
| **OPENROWSET()** |  |  |
| **OPENXML()** |  |  |
| **CHANGETABLE(CHANGES)** |  |  |
| **CHANGETABLE(VERSION)** |  |  |

### 05.03.10、安全函数(Security Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **::FN\_TRACE\_GETEVENTINFO()** |  |  |
| **::FN\_TRACE\_GETFILTERINFO()** |  |  |
| **::FN\_TRACE\_GETINFO()** |  |  |
| **::FN\_TRACE\_GETTABLE()** |  |  |
| **HAS\_DBACCESS()** |  |  |
| **ORIGINAL\_DB\_NAME()** |  |  |
| **IS\_MEMBER()** |  |  |
| **IS\_SRVROLEMEMBER()** |  |  |
| **SUSER\_SID()** |  |  |
| **SUSER\_SNAME()** |  |  |
| **USER** |  |  |
| **USER\_ID()** |  |  |
| **USER\_NAME()** |  |  |

### 05.03.11、字符串函数(String Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **ASCII()** |  |  |
| **CHAR()** |  |  |
| **CHARINDEX()** |  |  |
| **DIFFERENCE()** |  |  |
| **LEFT()** |  |  |
| **LEN()** |  |  |
| **LOWER()** |  |  |
| **LTRIM()** |  |  |
| **NCHAR()** |  |  |
| **PATINDEX()** |  |  |
| **QUOTENAME()** |  |  |
| **REPLACE()** |  |  |
| **REPLICATE()** |  |  |
| **REVERSE()** |  |  |
| **RIGHT()** |  |  |
| **RTRIM()** |  |  |
| **SOUNDEX()** |  |  |
| **SPACE()** |  |  |
| **STR()** |  |  |
| **STUFF()** |  |  |
| **SUBSTRING()** |  |  |
| **UNICODE()** |  |  |
| **UPPER()** |  |  |

### 05.03.12、系统统计函数(System Statistical Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **@@CONNECTIONS** |  |  |
| **@@CPU\_BUSY** |  |  |
| **::FN\_VIRTUALFILESTATS()** |  |  |
| **@@IDLE** |  |  |
| **@@IO\_BUSY** |  |  |
| **@@PACK\_RECEIVED** |  |  |
| **@@PACK\_SENT** |  |  |
| **@@PACKET\_ERRORS** |  |  |
| **@@TIMETICKS** |  |  |
| **@@TOTAL\_ERRORS** |  |  |
| **@@TOTAL\_READ** |  |  |
| **@@TOTAL\_WRITE** |  |  |

### 05.03.13、文本和图像函数(Text And Image Functions)

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **PATINDEX()** |  |  |
| **TEXTPTR()** |  |  |
| **TEXTVALID()** |  |  |

## 05.04、流程控制

|  |  |  |
| --- | --- | --- |
| **函数** | **功能** | **示例** |
| **BEGIN…END**  **单元语句** | 封装了多个T-SQL语句组合，将他们组成一个单元来处理，一般在条件查询或者循环等控制流程语句中，要查询满足特定条件的操作时会用到； | BEGIN  SELECT \* FROM StudentInfo  UPDATE StudentInfo SET Scope = 50  END |
| **IF…ELSE**  **判断语句** | 当满足某种特定条件的时候才执行的语句，通过做出判断，选择执行某条语句或语句块； | DECLARE @Scope INT  SELECT @Scope = Scope FROM StudentInfo where StuId = 1  IF @Scope >= 100  PRINT '满分'  ELSE  print '没学好' |
| **IF…EXISTS**  **检测语句** | 用来检测数据是否存在，当然了我们也可以通过COUNT(...)来实现，但是没有IF...EXISTS效果好，因为如果只要找到第一条匹配的数据的话，数据库就会停止检测； | IF EXISTS(SELECT \* FROM StudentInfo WHERE StuId = 1)  PRINT '存在'  ELSE  PRINT '不存在' |
| **CASE…WHEN**  **多分支判断语句** | CASE...WHEN结构提供比IF...ELSE结构更多的选择和判断机会，使用它可以很方便的实现多分支判断，从而避免IF...ELSE的语句嵌套使用； | SELECT Scope =  CASE  when(Scope < 60) THEN '不及格'  when(Scope = 60) THEN '及格'  when(Scope > 60 AND Scope < 80) THEN '良好'  ELSE '优秀'  END  FROM StudentInfo |
| **WHILE**  **循环语句** | 可以重复执行SQL语句或者要执行的语句块，只要指定的条件成立即可；BREAK命令让程序完全跳出循环语句，结束WHILE命令，CONTINUE是让命令继续返回执行； | DECLARE @i INT, @sum INT  SELECT @i = 1, @sum = 0  WHILE @i <= 100  BEGIN  SET @sum = @sum + @i  SET @i = @i+1  CONTINUE  END  PRINT '1 + 2 + 3 ...... 100 = '  PRINT @sum |
| **GOTO**  **跳转语句** | 使用GOTO语句可以改变程序的流程，让程序自动跳到我们要执行的程序行； | DECLARE @i INT, @sum INT  SELECT @i = 1, @sum = 0  WHILE @i <= 100  BEGIN  SET @sum = @sum + @i  SET @i = @i+1  GOTO GOTO1  END  PRINT '1 + 2 + 3 ...... 100 = '  PRINT @sum  -- GOTO;  GOTO1:  PRINT '跳出' |

## 05.05、注释

-- 单行注释;

/\*

多行注释;

\*/

# 06、系统数据类型的对照表(System Data Types)

<http://www.w3school.com.cn/sql/sql_datatypes.asp>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类型** | **T-SQL数据类型** | **C#数据类型** | **精度/范围** | **描述** |
| **精确数字**  **Exact Numerics** | **bit** | **Boolean** | 0 / 1 | 0 = false，1 = true； |
| **tinyint** | **Byte** | 0 ~ 255 | 存储大小为1字节； |
| **smallint** | **Int16** | -2^15(-32768) ~ 2^15 - 1(32767) | 存储大小为2个字节； |
| **int** | **Int32** | -2^31(-2147483648) ~ 2^31 - 1(2147483647) | 存储大小为4个字节； |
| **bigint** | **Int64** | -2^63(-9223372036854775808) ~ 2^63 - 1(9223372036854775807) | 存储大小为8个字节； |
| **numeric** | **Decimal** | -10^38 + 1 ~ 10^38 – 1 | 固定精度和小数位的数字； |
| **decimal** | **Decimal** | -10^38 + 1 ~ 10^38 – 1 | 固定精度和小数位的数字； |
| **smallmoney** | **Decimal** | -214748.3648 ~ 214748.3647 | 精确到货币单位的千分之十； |
| **money** | **Decimal** | -2^63(-922337203685477.5808) ~ 2^63 - 1(922337203685477.5807) | 精确到货币单位的千分之十； |
| **近似数字(浮点)**  **Approximate Numerics** | **float** | **Double** | -1.79E + 308 ~ 1.79E + 308 | 浮点精度数字； |
| **real** | **Single** | -3.40E + 38 ~ 3.40E + 38 | 浮点精度数字； |
| **日期和时间**  **Date And Time** | **datetime** | **DateTime** | 1753年1月1日 ~ 9999年12月31日 | 精确到百分之三秒(或3.33毫秒)； |
| **smalldatetime** | **DateTime** | 1900年1月1日 ~ 2079年6月6日 | 精确到分钟； |
| **date** | **DateTime** |  |  |
| **time** | **DateTime** |  |  |
| **datetimeoffset** | **DateTime** |  |  |
| **datetime2** | **DateTime** |  |  |
| **字符串**  **Character Strings** | **char** | **String** | 0 ~ 8000字符 | 固定长度的非Unicode字符数据； |
| **varchar** | **String** | 0 ~ 8000字符 | 可变长度的非Unicode字符数据； |
| **text** | **String** | 0 ~ 2^31 - 1(2147483647) 字符 | 可变长度的非Unicode字符数据； |
| **Unicode字符串**  **Unicode Character Strings** | **nchar** | **String** | 0 ~ 2^31 - 1(2147483647) 字符 | 固定长度的Unicode字符数据； |
| **nvarchar** | **String** | 0 ~ 4000字符 | 可变长度的Unicode字符数据； |
| **ntext** | **String** | 0 ~ 2^30 - 1(1073741823) 字符 | 可变长度的Unicode字符数据； |
| **二进制**  **Binary Strings** | **binary** | **Byte[]** | 0 ~ 8000字节 | 固定长度的二进制数据 |
| **varbinary** | **Byte[]** | 0 ~ 8000字节 | 可变长度的二进制数据 |
| **image** | **Byte[]** | 0 ~ 2^31 - 1(2147483647) 字节 | 可变长度的二进制数据 |
| **其他**  **Other Data Types** | **sql\_variant** | **Object** |  | 各种数据类型但不包括text/ntext/image/timestamp/sql\_variant； |
| **timestamp** | **DateTime** |  | 数据库范围的唯一数字，每次更新行时刷新； |
| **uniqueidentifier** | **Guid** |  | 全局唯一标识符(GUID)； |
| **xml** |  |  |  |
| **CLR数据类型**  **CLR Data Types** | **hierarchyid** |  |  |  |
| **空间数据类型**  **Spatial Data Types** | **geometry** |  |  |  |
| **geography** |  |  |  |