

大对象存储在数据行外时 CREATE TABLE T() ON [PRIMARY] TEXTIMAGE_ON my_filegroup

显式类型转换 CAST (代码 as varchar(10)); CONVERT (varchar(10), GETDATE(), 110) AS 110

隐式类型转换 Float 和 int, Float 和 char

局部变量 DECLARE @variable Type; SET @variable=[表达式|子查询]; SELECT @variable=[表达式|字段名] FROM WHERE

条件分支 IF 表达式 [BEGIN ... END] ELSE IF [] ELSE

CASE 表达式 WHEN 常量值 THEN [] ELSE [] END

CASE WHEN 条件表达式 THEN [] ELSE END CASE 是表达式有返回值

DATEPART(year|quarter, getdate())整数|DATENAME 字符串|DATEADD(day, num, getdate()) DATEDIFF(week, date1, date2)

字符串 LOWER; UPPER; RTRIM 截断右部空格; STR 数值转字符串; SUBSTRING(str, st, len); CHARINDEX(str1, str2)1 在 2

中的起始位置; PATINDEX('%pattern%', str)起始位置, 没有则返回 0; REPLACE(str1, str2, str3)在 1 中把 2 替换为 3;

CREATE TRIGGER name ON table AFTER INSERT AS BEGIN [] END

CREATE PROC name @variable int=3 AS [BEGIN (RETURN) END] GO EXEC name

SELECT * INTO #临时表 FROM #本地 ##全局

CREATE FUNCTION name (@variable type) RETURNS type AS BEGIN [RETURN sth.] END

CREATE FUNCTION name () RETURNS TABLE AS RETURN(SELECT ...)

CREATE FUNCTION name () RETURNS @var TABLE(定义) AS BEGIN [RETURN @var] END

连接查询: SELECT ... FROM T1 INNER JOIN T2 ON T1.A=T2.A ||TOP 查询: SELECT TOP (表达式) PERCENT * FROM

分区提取 TOP N: SELECT ROW_NUMBER() OVER(PARTITION BY Gamedate ORDER BY PDpoint DESC) AS row_num, * INTO

#t2 FROM #t1|SELECT * FROM #t2 WHERE row_num<5 ORDER BY 排名 RANK() OVER() 密集排名 DENSE_RANK() OVER()

数据分页: WITH SN_T AS (SELECT ROW_NUMBER() OVER (ORDER BY A) rn, * from T) SELECT * FROM T RN BETWEEN

多页访问 SELECT ROW_NUMBER() OVER(ORDER BY A) row_num, * INTO #T1 FROM T

CREATE CLUSTERED INDEX idx_rn ON #T1(row_num)

SELECT * FROM #T WHERE row_num BETWEEN @PageSize*(@Page_Num-1)+1 and ...

累计聚合: SELECT SUM(point) OVER(PARTITION BY Pid ORDER BY Gameid ROWS BETWEEN [UNBOUNDED|num]

[PRECEDING|FOLLOWING] AND CURRENT ROW) AS A FROM PlayerData ORDER BY Pid, Gameid

不被截断 INSERT INTO T() SELECT REPLICATE(CONVERT(VARCHAR(MAX), 's'), 9000)

读文件 INSERT INTO T() SELECT * FROM OPENROWSET(BULK ' ', SINGLE_C/BLOB) AS

启用全文索引 SP_FULLTEXT_DATABASE 'ENABLE' GO RECONFIGURE GO

检索 WHERE CONTAINS(列, 'FORMSOF() or '); 加权 FROM CONTAINSTABLE(表,列,'isabout("content" weight(0.3),)', 返回记

录数) as t2; Freetext(列, '短语/词语/句子'); ALTER FULLTEXT INDEX ON 表 SET STOPLIST OFF/SYSTEM

短语加密 ENCRYPTBYPASSPHRASE(@pwd, '123')/DECRYPTBYPASSPHRASE(@pwd, 列)

对称密钥 CREATE SYMMETRIC KEY Gamekey WITH ALGORITHM=AES_256 ENCRYPTION BY PASSWORD='123' ||

打开/关闭密钥 OPEN/CLOSE SYMMETRIC KEY Gamekey DECRYPTION BY

PASSWORD='123' || ENCRYPTBYKEY(KEY_GUID('Gamekey'), @str) || CONVERT(VARCHAR(MAX), DECRYPTBYKEY(列))

非对称密钥 CREATE ASYMMETRIC KEY WITH ALGORITHM=RSA_2048 ENCRYPTION BY

PASSWORD='pwd' || ENCRYPTBYASYMKEY(ASYMKEY_ID(' '), @str) || DECRYPTBYASYMKEY(ASYMKEY_ID(' '), 列, N'pwd')

JAVA 连接 || import java.sql.*; import java.util.Scanner; public class myclass { public static void main(String[] args) { String

url='jdbc:sqlserver://localhost:1433;DatabaseName=NBA;trustServerCertificate=true'; String user=''; String pwd=''; String

driverName='com.microsoft.sqlserver.jdbc.SQLServerDriver'; Connection my_conn=null; try { Class.forName(driverName); my_conn=Driver

Manager.getConnection(url, user, pwd); System.out.println(''); } catch (Exception e) { e.printStackTrace(); } myclass t=new

myclass(); t.data_sel(my_conn); }

Public void data_sel(Connection conn) { String sql = '?'; Scanner sc1=new Scanner(System.in); String coach=sc1.next(); sc1.close(); try

{ PreparedStatement ps=conn.prepareStatement(sql); ps.setString(1, coach); ResultSet rs=ps.executeQuery/ps.executeUpdate(); } }

关系数据转 XML 数据 || FOR XML RAW || FOR XML AUTO || ← 每行为元素, 列为属性 FOR XML PATH 每列为 <row> 的子元素

XML 转关系 || DECLARE @docHandle int; Exec sp_xml_preparedocument @docHandle OUTPUT, @mydoc; SELECT * into TB1 FROM OPENXML(@docHandle, 'products/product', 1 属性值/2 元素值/3=1+2) WITH (Category NVARCHAR(50))

创建设备 EXEC sp_addumpdevice 'disk', 'my_bkd', 'E:\buaa\demo.bak' || 备份 BACKUP DATABASE NBA TO my_bkd WITH

INIT(完整备份) | DIFFERENTIAL, NO INIT(差异备份), NAME=' ' || BACKUP LOG NBA TO my_bkd WITH NO INIT, NAME=' '

恢复 RESTORE DATABASE NBA FROM my_bkd WITH RESTRICTED_USER, FILE=完整 1 差异 2 日志 3, NORECOVERY

游标 DECLARE CURSOR .. FOR (SELECT) | OPEN | FETCH NEXT FROM .. INTO .. | WHILE @@FETCH_STATUS=0 | CLOSE | DEAL