Funciones de SQL

Características principales

Las que son definidas por el usuario

Las que son propiamente del gestor

funciones DE SQL

ASCII

Devuelve el valor de código ASCII del carácter situado mas a la izquierda de una expresión de caracteres

Ejemplo

Select ascii(‘virginia’)

Select ascii(‘@correo’)

CHAR

Devuelve el carácter según el código ASCII correspondiente a un número entero

Select CHAR (86) + CHAR(64)

Select CHAR(118)

Select CHAR(270)

LEN

Devuelve el número de caracteres de la expresión de cadena especificada, excluidos los espacios en blanco finales

Select LEN(‘ virginia ‘’)

Select len(‘prueba la función len ’)

LTRIM

Devuelve una cadena tras quitar todos los espacios iniciales en blanco.

Select LTRIM(‘ prueba de la función LTRIM ’)

RTRIM

Devuelve una cadena tras quitar todos los espacios finales en blanco

Ejemplo

SELECT RTRIM('PRUEBA DE LA FUNCION RTRIM ')

SELECT \*, RTRIM(NOM\_E) AS INICIALES,

RTRIM(NOM\_E) AS FINALES

FROM ESCUELA

LEFT

Devuelve la parte izquierda de una cadena de caracteres con el numero de caracteres especificado

SELECT LEFT('VIRGINIA',5)

SELECT LEFT('UNIVERSIDAD',3)

RIGHT

Devuelve la parte derecha de una cadena de caracteres con el numero de caracteres especificado

Select right(‘virginia’,3)

SUBSTRING

Devuelve parte de una expresion de caracteres, binaria, de texto o de imagen,

Select substring(‘virginia’,2,4)

Select substring(‘virginia’,2,3)

LOWER

Devuelce una cadena despues de convertir en minusculas los caracteres en mayusculas

Select lower(‘virginia’)

UPPER

Devuelve una cadena despues de convertir en mayusculas los datos de caracteres en minusculas.

Select UPPER(‘virgina’)

REPLACE

Reemplaza todas las apariciones de un valor de cadena especificado por otro valor de cadena

Select replace(‘pruebas de reemplazo,’e’,’i’)

REPLICATE

Repita una expresion de caracteres un numero de especificado de veces.

Select replicate(‘virginia’,5)

REVERSE

Devuelve una expresion de caracteres invertida.

Select reverse(‘anita’)

fUNCIONES DE TIPO FECHA

GETDATE()

MUESTRA LA FECHA ACTUAL

SELECT GETDATE()

DATEADD

EJECUTA OPERACIONES MATEMATICAS CON VALORES DE FECHA

SINTAXIS

DATEADD(CODIGO PARTE DE LA FECHA, NUMERO, FECHA SQL)

CODIGO FECHAS SQL

Año yy,yyyy

Trimestre qq,q

Mes mm,m

Dia del año dy,y

Semana wk,ww

Horas Hh

Minuto mi,n

Segundo ss,s

Milisegundo Ms

DATEDIFF

DEVUELVE EL NUMERO DE UNIDADES DE FECHA ENTRE DOS FECHAS ESPECIFICADAS

SINTAXIS

DATEDIFF

(CODIGO PARTE DE LA FECHA, FECHA DE INICIO, FECHA FINAL SQL)

DATENAME

DEVUELVE LA APRTE ESPECIFICADA DE LA FECHA COMO CADENA DE CARACTERES

SINTAXIS

DATENAME(CODIGO PARTE DE LA FECHA, FECHA SQL)

DAY

DEVUELVE EL DIA ASOCIADO CON LA EXPRESION DE UNA FECHA COMO ENTERO

SINTAXIS

DAY(FECHA\_SQL)

MONTH

DEVUELVE EL MES ASOCIADO CON LA EXPRESION DE UNA FECHA COMO ENTERO.

SINTAXIS

MONTH(FECHA\_SQL)

YEAR

FEVUELVE EL AÑO ASOCIADO CON LA EXPRESION DE UNA FECHA COMO ENTERO

Un desencadenador (o trigger) es una clase especial de procedimiento almacenado que se ejecuta automáticamente cuando se produce un evento en el servidor de bases de datos

SQL Server permite crear varios desencadenadores para una instrucción especifica.

Según el tipo de evento que los desencadena se clasifican en.

Ventaja es no se tiene que estar ejecutando, esperando a que algo suceda, hacer determinada accion, se dispara.

Desencadenadores DML

Desencadenadores DDL

Desencadenadores LOGON(inicios de sesiones)

**Los desencadenadores DML** se ejecutan cuando un usuario intenta modificar datos mediante un evento de lenguaje de manipulacion de datos(DML).

Los eventos DML son instrucciones INSERT, UPDATe o DELETE de una tablao vista

**Los desencadenadores DDL** se ejecutan en respuesta a una variedad de eventos de lenguaje de definicion de datos(DDL.

Estos eventos corresponde principalmente a insturcciones CREATE, ALTER y DROP de TRANSACT SQL ya determinados procedimientos almacenados del sistema que ejecutan operaciones de tipo DDL

**Los desencadenadores logon** se activan en respuesta al evento LOGON que se genera cuando se establece la sesion de un usuario

SINTAXIS

Create trigger nombre (crear el nombre de disparador)

On tabla (especifica el nombre de la tabla que se va usar para el disparador)

For (sirve para indicar al disparador cuando se va a disparar)(tipo de instrucción(evento))

As (empieza el cuerpo de disparador)

Begin

Cuerpo del trigger ()

End

vARIABLES

SINTAXIS:

Declare @nomVar tipo

Asignar valores a variable

Representa el valor de un campo el contenido del campo

Tablas inserted son tablas virtuales o tablas fantasmas

Select variable = tabla/alias.campo from tabla

If(condicion(es))

Begin

Instrucciones

End

Una variable debe de llevar el signo de @

Declare @A Varchar(70)

Declare @pr float

Select @A=I.nomprod

@pr=I.precio\_comp

From Inserted I

Transacion: cualquier manipulacion que se le quiere realizar a la base de datos

Puede terminar terminar solamente de dos formas

Termina con éxito o termina con fracaso

Una trasaccion jamas se debe dejarlo a medias

Consistencia de base de datos, se encarga de que la base de datos quede completa y correcta

Cuando una transacicon termnia con éxito se termina con un commit

Cuando una transaccion termina fracaso se termina con rollback

Rollback

Commit

Altertable

Sirve para agregar nuevos campos a la tabla

Alter table nombre\_tabla

Add nombreCampo tipodato

Exist una instruccion que sirve si existe algo

Proceidmientos almacenados

Son elementos que se pueden programar para hacer determinadas acciones

NO se dispara de manera automatica

Se tiene que ejecutar oprogramar para que sea ejecutable

Puede ser programado para multiples cosas

Seconocen SP

Los procedimientos almacenados son al nivel DDL

SINTAXIS:

CREATE procedure nombre parametros

Para ejecutar un procedimiento almacenado es con

EJECUTAR

EXECUTE NOMBRE DEL PROCEDIMIENTO

NOMBRE DEL PROCEDIMIENTO