

# LAS NUEVAS CARACTERISTICAS DEL ESTANDAR ISO/IEC 9075:2008

Huichi Contreras, Franklin Carlos (2016054948)

*Tacna, Perú*

---

## Abstract

The ISO / IEC 9075 defines the structure and functions that we can use in the SQL-oriented database language where the structure of the Query language is described, in order to be able to make SQL statements and that are subsequently carried out successfully being these independent to the previously established or recently established SQL statements. That is why the need to improve and renew has made that from time to time new features are established in the SQL database query language, which is why this article will detail the new features that ISO / IEC 9075: 2008 brought us.

---

## 1. Resumen

El ISO / IEC 9075 nos define la estructura y funciones que podemos utilizar en el lenguaje SQL orientado a base de datos donde se describe la estructura del lenguaje Query, con el fin de poder hacer declaraciones SQL y que sean posteriormente realizadas con éxito siendo estos independientes a las declaraciones SQL anteriormente establecidas o recientemente establecidas. Es por ello que la necesidad de mejorar y renovar han hecho que cada cierto tiempo se vayan estableciendo nuevas características al SQL database query language, es por ello que en este artículo se detallara las nuevas características que nos trajo el ISO / IEC 9075 : 2008.

## 2. Introduccion

SQL es un lenguaje de consulta textual declarativo para bases de datos relacionales. Combina consultas, actualizaciones e instrucciones administrativas en un solo idioma el idioma SQL. Las instrucciones administrativas

permiten cambiar las diferentes propiedades de una base de datos mientras se está ejecutando.

### 3. Marco Teorico

#### 3.1. NUEVAS CARACTERISTICAS DEL ISO / IEC 9075:2008

Estas son las principales características que se pueden encontrar en la versión SQL 2008 :

- Declaración del término MERGE este comando proporciona la capacidad de INSERT programáticamente datos si no existe o ACTUALIZAR los datos.
- Se introdujo la nueva declaración Truncate Table.
- La declaración de variables permite una inicialización similar a la que hacemos en C++ (Declarábamos y luego lo utilizábamos). Ahora, en lugar de escribir dos declaraciones, podemos escribir una sola declaración (Ejemplo: DECLARE @COUNT INT =100).

```
DECLARE @COUNT INT =100
```

- La declaración INSERT INTO ahora se puede agregar como cadena separado por una coma luego de declarar la sentencia de VALUES (Ej.: INSERT INTO Letras VALUES ((1,A), (2,B), (3,C) ) )

```
INSERT INTO dbo.Employee  
VALUES(1, 'Basavaraj' ) ,  
      (2, 'Shashank' ) ,  
      (3, 'Monty')
```

- Se mejoro al momento de declarar la llave When donde se agregó como ayuda una separación por comas en una expresión que contenga la sentencia CASE.

- Se introdujo Sparse Column que es una nueva característica más introducida en SQL SERVER 2008. Almacenar un valor nulo en una columna dispersa no ocupa espacio, pero almacenar un valor no nulo en una columna dispersa ocupa 4 bytes de espacio adicional que las columnas no dispersas de el mismo tipo de datos.
- Se agrego nuevos tipos de datos de fecha y hora como el Date, Time, DateTime2, etc.
- Tambien se introdujo los operadores de asignacion aritmeticos como los siguientes:

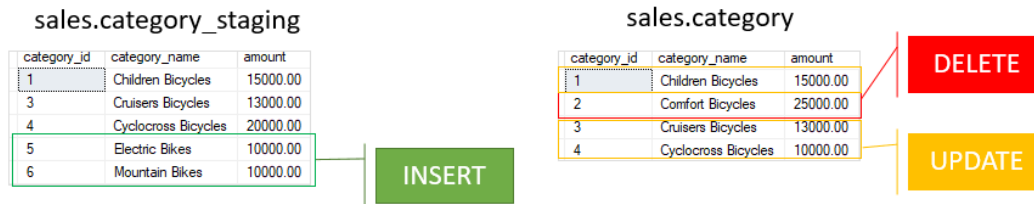
Operator	Usage	Description
+=	<b>SET</b> @x+=@y	Same <b>as</b> : <b>SET</b> @x = @x + @y
-=	<b>SET</b> @x-=@y	Same <b>as</b> : <b>SET</b> @x = @x - @y
*=	<b>SET</b> @x*=@y	Same <b>as</b> : <b>SET</b> @x = @x * @y
/=	<b>SET</b> @x/=@y	Same <b>as</b> : <b>SET</b> @x = @x / @y
%=	<b>SET</b> @x%=@y	Same <b>as</b> : <b>SET</b> @x = @x % @y

#### 4. Analisis

Analizaremos una de las sentencias que mas causo impacto la instrucción Merge. La instrucción Merge fue lanzada en la versión 2008 con esta función podemos realizar múltiples operaciones como el INSERT, UPDATE y DELETE en los datos de la tabla de destino según sea visto en la tabla de origen y la condición que se da para llevar a cabo una unión entre ellos. Esta característica es útil al momento de querer sincronizar datos de la tabla de destino con los datos de la tabla de origen. En versiones anteriores para poder lograr una sincronización, tendríamos que haber escaneado las tablas de origen y destino repetidas veces, pero con la instrucción Merge podemos lograr esto con una sola instrucción y además con una sola búsqueda en la tablas de origen y destino.

Por ejemplo en la imagen mostrada podemos ver que la tabla sales.category.staging y sales.category tienen los mismos campos y los mismos datos, con excepción de que tiene algunas pequeñas diferencias.

El código mostrado podemos ver que usamos la sentencia MERGE y vamos a alterar la tabla sales.category y usaremos la tabla sales.category.staging,



```

MERGE sales.category t
  USING sales.category_staging s
 ON (s.category_id = t.category_id)
WHEN MATCHED
  THEN UPDATE SET
    t.category_name = s.category_name,
    t.amount = s.amount
WHEN NOT MATCHED BY TARGET
  THEN INSERT (category_id, category_name, amount)
    VALUES (s.category_id, s.category_name, s.amount)
WHEN NOT MATCHED BY SOURCE
  THEN DELETE;

```

estableceremos diciendo que se asemejen en su category.id ya que ambos campos lo tienen. Entonces cuando encuentren igualdades (ósea que el id de la tabla 1 sea 1 y el id de la tabla 2 sea 1) se procederá a actualizar los campos del nombre y el monto a cómo está la tabla sales.category.staging. En caso de que no encuentre igualdad en los ID entonces se insertara a los campos de sales.category los valores de la tabla sales.category.staging. Y por último como la tabla sales.category.staging no tiene el id 2 se borra quedando el resultado como es visto en la imagen.

category_id	category_name	amount
1	Children Bicycles	15000.00
3	Cruisers Bicycles	13000.00
4	Cyclocross Bicycles	20000.00
5	Electric Bikes	10000.00
6	Mountain Bikes	10000.00

## 5. Conclusion

La realización de la investigación acerca de las novedades que nos trajo el ISO/IEC 9075:2008 me llevaron a la conclusión de que pude aprender nuevas sentencias que fueron establecidas para resolver problemas al mantenimiento de la base de datos. Por ejemplo, el MERGE resolvía el problema que al momento de ACTUALIZAR, ELIMINAR Y AGREGAR datos de una tabla a otra vinculadas era mucho más difíciles hacerlo manualmente sin esta sentencia, o también por el lado de seguridad para poder ingresar datos exclusivos como fechas usaban los varchar esto lo arreglaron usando nuevos tipos de variables enfocadas al tiempo mejorando el filtro de entradas que se le daba a estos datos. Esto me hace pensar que todavía nos falta mucho para poder redefinir una estructura SQL finalizada o terminada, ya que cada vez que van avanzando la tecnología y los datos el manejo de la información necesita ser cada vez mejor manejada y mejor optimizada todo ello derivado a los datos que recaudemos de nuestros usuarios y para poder tener más formas de poder manejar las bases de datos.

## Referencias

- [1] Biradar, B. (2011). New features in sql server 2008. Recuperado de <https://sqlhints.com/2011/09/11/new-features-in-sql-server-2008/>. Accedido 18-08-2019.
  - [2] Boyd, C. (2007). Sql server 2008 features, function and value. Recuperado de <https://www.mssqltips.com/sqlservertip/1313/sql-server-2008-features-function-and-value/>. Accedido 17-08-2019.
  - [3] Chamberlin, D. D. (2012). Early history of sql. Recuperado de <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6359709>. Accedido 17-08-2019.
  - [4] Microsoft (2012). What's new (sql server 2008). Recuperado de [https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/bb500435\(v=sql.100\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/bb500435(v=sql.100)). Accedido 15-08-2019.
  - [5] Rodríguez, V. J. V. (ne). Administración de base de datos con sql server 2008. Recuperado de <http://www.v-espino.com/che-ma/daw1/tutoriales/SQLServer.pdf>. Accedido 18-06-2019.
- [1] [2] [4] [3] [5]