

# **NATURALEZA DE LOS DATOS**

**AUTOR: MAX JOSÉ BERMÚDEZ LEÓN**

**DICIEMBRE: 2020**



## Introducción

Las bases de datos son un elemento fundamental en el entorno informático hoy en día y tienen aplicación en la práctica totalidad de campos. Concebidas con un propósito general, son de utilidad para toda disciplina o área de aplicación en la que exista una necesidad de gestionar datos, tanto más cuanto más voluminosos sean estos. En nuestro ámbito particular de los SIG, los datos son cada día más voluminosos, debido no solo a una mayor cantidad de información, sino también a una mayor precisión en esta, la cual implica un mayor volumen de datos. Además, presentan otra serie de características (uso múltiple, necesidad de acceso eficiente para análisis, necesidad de indexación, etc.), haciendo todas ellas que sea recomendable el uso de bases de datos y tecnologías específicas para su manejo.



## Tabla de contenido

Introducción.....	1
Naturaleza de los Datos en SQL .....	3
Introducción al lenguaje SQL .....	3
Sistema de bases de datos Relacionales .....	3
Utilización del lenguaje .....	4
Estandarización del SQL .....	4
Características del Lenguaje SQL .....	5
Sentencias SQL .....	5
Tipos de Sentencias .....	5
Tablas.....	8
Tipos de datos.....	9
Tipo de dato Carácter y Cadena de Caracteres.....	12
Tipo de datos Booleanos.....	14
Tipos de datos Numéricos .....	15
Tipos de dato Fecha y Hora .....	18
Tipos de datos de cadenas y caracteres Unicode SQL .....	22
Tipos de datos binarios SQL .....	23
Conclusiones y recomendaciones .....	25
Referencias bibliográficas.....	26

## Naturaleza de los Datos en SQL

### Introducción al lenguaje SQL

El lenguaje SQL permite la comunicación con los Sistemas de Bases de Datos Relacionales (SGBD). La palabra SQL está formada por las iniciales de STRUCTURED QUERY LANGUAGE (Lenguaje de Consulta Estructurado). Esta definición inicial no describe completamente las posibilidades del lenguaje, ya que con él se hacen todas las operaciones necesarias para la gestión del SGBD (actualizaciones de datos, definición de objetos, operaciones de control y consultas de datos). Una de las características más importantes de SQL es que sus sentencias pueden manejar conjuntos de registros. Esto confiere al lenguaje una gran potencia, consiguiéndose una alta productividad.

**Un Sistema de Gestión de Base de Datos relacional (SGBD) es un conjunto de programas que se encarga de gestionar los datos almacenados en la Base de Datos. Estos sistemas se caracterizan por: Los datos se presentan a los usuarios en forma de tablas, formadas por filas y columnas**

### Sistema de bases de datos Relacionales

- Un Sistema de Gestión de Base de Datos relacional (SGBD) es un conjunto de programas que se encarga de gestionar los datos almacenados en la Base de Datos. Estos sistemas se caracterizan por: Los datos se presentan a los usuarios en forma de tablas, formadas por filas y columnas.
- Todas las operaciones se realizan con sentencias SQL.
- Existe una independencia entre el almacenamiento físico de los datos y la estructura lógica de la base de datos (tablas).

El SGBD coordina todas las peticiones realizadas por los usuarios, asegurando en todo momento la integridad de los datos. Además, el SGBD tiene un conjunto de tablas predefinidas que forman el Diccionario de Datos o Catálogo. En estas tablas se guardan las

definiciones de los objetos/estructuras (tablas, vistas, índices, etc.) que componen la base de datos. Estas tablas son mantenidas automáticamente por el SGBD y pueden ser consultadas por los usuarios.

## Utilización del lenguaje

Las sentencias SQL pueden escribirse directamente en un terminal interactivo en el cual se reciben los resultados de las operaciones pedidas. También se utilizan sentencias SQL dentro de programas de tercera y cuarta generación (COBOL, FORTRAN, C, PL/SQL, etc.) que tratan datos almacenados en bases de datos relacionales.

## Estandarización del SQL

Los institutos American National Standards Institute (ANSI) y International Standards Organisation (ISO) han aceptado SQL como el lenguaje de las bases de datos relacionales.

Las últimas normas sobre SQL publicadas llamadas SQL92 o SQL2 son:

- ANSI X3.135-1992, "Database Language SQL".
- ISO/IEC 9075:1992, "Database Language SQL".

Aunque los sistemas de bases de datos de la industria no cumplen con todas estas especificaciones, todos están realizados basándose en ellas, esto hace que las diferencias más importantes aparezcan en opciones y en funciones de las sentencias del lenguaje.

## Características del Lenguaje SQL

El lenguaje SQL sirve para manejar las bases de datos, es un lenguaje de texto plano, que podemos usar desde cualquier editor de textos. Nos permite acceder y manipular las bases de datos. Para usarlo con páginas web debemos usarlo conjuntamente con otros lenguajes. Con SQL podemos hacer entre otras las siguientes operaciones en el programa que gestiona las bases de datos:

- Crear nuevas bases de datos.
- Crear nuevas tablas en una base de datos.
- Crear nuevos registros en las tablas de una base de datos
- Crear tablas de consulta en una base de datos.
- Borrar tablas o bases de datos.
- Borrar registros.
- Cambiar uno o varios datos de un registro.
- Mostrar los datos de una tabla o consulta.
- Buscar y mostrar sólo algunos datos de una tabla o consulta, poniendo una serie de condiciones para la búsqueda.

Es decir, el lenguaje SQL permite gestionar una base de datos en su totalidad.

## Sentencias SQL

Una sentencia SQL es una cadena de caracteres que se envía al SGBD para su ejecución. Contiene palabras del lenguaje, nombres de tablas y columnas, constantes y signos delimitadores. Esta cadena de caracteres se compila automáticamente, generándose un procedimiento de ejecución que realiza la operación deseada. Si la sentencia es errónea o incompleta el SGBD genera un mensaje de error.

### Tipos de Sentencias

Las sentencias SQL se dividen en las siguientes categorías o grupos:



- Sentencias de manipulación de datos (Data Manipulation Language DML).
- Sentencias de definición de datos (Data Definition Language DDL).
- Sentencias de control de control (Data Control Language DCL).
- Sentencias de Control de Transacción (Transaction Control Language).

DDL: Lenguaje de definición de Datos: (Data Definition Languaje) Entran aquí las sentencias que definen y crean los objetos que soportan la base de datos (creación o supresión de bases, creación o supresión de tablas, relaciones entre tablas, claves en las tablas, etc.

DML: Lenguaje de Manipulación de Datos: (Data Management Languaje) Entran aquí las sentencias para manejar los datos almacenados en las tablas, a nivel de campos (columnas) o registros (filas). Por ejemplo, crear, cambiar o consultar registros o sus datos.

DCL: Lenguaje de Control de Datos: (Data Control Languaje) Entran aquí las sentencias para controlar las funciones de administración y control de las bases de datos.

TCL: Lenguaje de control de Transacciones: (Transaction Control Language) Se utilizan para controlar el procesamiento de transacciones en una base de datos. Una transacción es una unidad lógica de trabajo que comprende una o más sentencias SQL, por lo general un grupo de Data Manipulation Language (DML).

Estos dos lenguajes no son lenguajes en sí mismos, sino que es una forma de clasificar las sentencias de lenguaje SQL en función de su cometido.

La diferencia principal reside en que el DDL crea objetos en la base de datos y sus efectos se pueden ver en el diccionario de la base de datos; mientras que el DML es el que permite consultar, insertar, modificar y eliminar la información almacenada en los objetos de la base de datos.

Cuando se ejecutan las sentencias DDL de SQL, el SGBD confirma la transacción actual antes y después de cada una de las sentencias DDL. En cambio, las sentencias DML no

llevan implícito el commit y se pueden deshacer.

Existe pues un problema al mezclar sentencias DML con DDL, ya que estas últimas pueden confirmar las primeras de manera involuntaria e implícita, lo que en ocasiones puede ser un problema.

A continuación, se presenta una tabla con las sentencias SQL más comunes, clasificadas según el lenguaje al que pertenecen.

<b><i>Sentencia DDL</i></b>	<b><i>Objetivo</i></b>
Alter procedure	Recompilar un procedimiento almacenado.
Alter Table	Añadir o redefinir una columna, modificar la asignación de almacenamiento.
Analyze	Recoger estadísticas de rendimiento sobre los objetos de la BD para utilizarlas en el optimizador basado en costes.
Create Table	Crear una tabla.
Create Index	Crear un índice.
Drop Table	Eliminar una tabla.
Drop Index	Eliminar un índice.
<b><i>Sentencia DCL</i></b>	<b><i>Objetivo</i></b>
Grant	Conceder privilegios o papeles, roles, a un usuario o a otro rol.
Truncate	Eliminar todas las filas de una tabla.
Revoke	Retirar los privilegios de un usuario o rol de la base de datos.
<b><i>Sentencia DML</i></b>	<b><i>Objetivo</i></b>



Insert	Añadir filas de datos a una tabla.
Delete	Eliminar filas de datos de una tabla.
Update	Modificar los datos de una tabla.
Select	Recuperar datos de una tabla.
<b>Sentencia TCL</b>	<b>Objetivo</b>
Commit	Confirmar como permanentes las modificaciones realizadas.
SAVEPOINT	Identifica un punto de una transacción a la que más tarde se puede volver.
SET TRANSACTION	Cambia las opciones de transacción como nivel de aislamiento y que segmento de cancelación utiliza.
Rollback	Deshacer todas las modificaciones realizadas desde la última confirmación.

## Tablas

Todos los datos almacenados en las Bases de Datos relacionales están dispuestos en estructuras llamadas tablas. Cada tabla se identifica con un nombre y está compuesta por un conjunto de columnas. A su vez cada columna se define con un nombre, un tipo de dato (alfanumérico, numérico, fecha, etc.) y una longitud. Las columnas se identifican por su nombre, no por su posición dentro de la tabla. Por tanto, el orden de las columnas dentro de la tabla no tiene importancia. La información dentro de la tabla se guarda en filas, conteniendo datos de las columnas de la tabla. No es significativo el orden en el que se almacenan las filas.

En cada intersección de una fila y una columna se almacena un solo valor. Aceptándose el valor NULL (ausencia de valor) que no ocupa espacio en la base de datos.

Los usuarios que crean las tablas son los únicos que pueden trabajar inicialmente con los datos almacenadas en ellas. Por tanto, según los criterios de confidencialidad fijados, es

necesario autorizar el acceso al resto de los usuarios

## Tipos de datos

En la definición de columnas de las tablas se utilizan fundamentalmente tres tipos de datos:

**Columnas numéricas.** Contienen números enteros, decimales y en coma flotante. La norma ANSI define tipos diferentes para cada uno de estos números: SMALLINT para enteros pequeños, INTEGER para enteros grandes, NUMERIC y DECIMAL para números con parte entera y parte fraccionaria y FLOAT, DOUBLE PRECISION Y REAL para números en formato en coma flotante. En el manual se utiliza el tipo NUMBER (p, s) de Oracle con el que se pueden implementar los tipos anteriores. En la definición NUMBER (p, s), el entero 'p' se llama precisión y es el número de total de dígitos (parte entera y parte fraccionaria). El entero 's', es el número de dígitos que puede tener la parte fraccionaria.

**Columnas alfanuméricas.** Contienen cadenas de caracteres de longitud fija y de longitud variable. La norma ANSI define tipos de datos de longitud fija con CHARACTER, CHAR, NATIONAL CHARACTER, NCHAR, etc. y tipos de longitud variable con CHARACTER VARYING, CHAR VARYING, NATIONAL CHARACTER VARYING, NCHAR VARYING, etc. En el manual se utiliza en la definición de este tipo de columnas el tipo VARCHAR(n) de Oracle, que permite cadenas de caracteres de longitud variable con valor máximo igual al entero 'n'. Las columnas alfanuméricas de longitud variable se caracterizan principalmente porque el SGBD requiere un espacio de almacenamiento igual a la longitud real del dato no a la longitud máxima de definición de la columna.

**Columnas de fechas.** Estas columnas contienen información de datos temporales (año, mes, día, hora, minuto y segundo). Este tipo de datos no está definido en la norma ANSI, por lo cual cada fabricante lo implementa de forma diferente. Se utilizan los tipos: TIME para contener horas, TIMESTAMP para contener instantes en el tiempo y DATE para contener días. Existen otros tipos de datos que permiten guardar información de gran

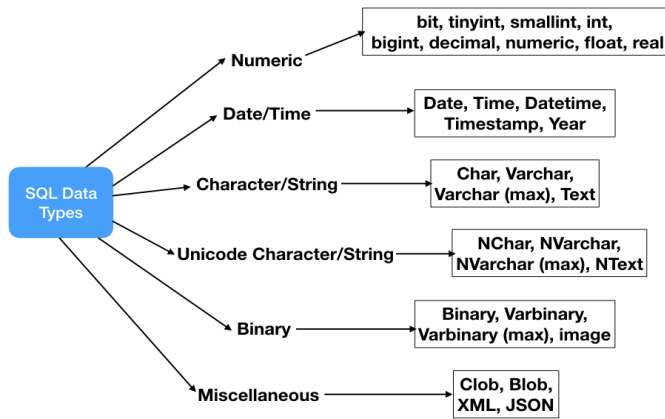
tamaño(gigabytes) tanto de tipo binario como de tipo alfanumérico. Estos tipos tampoco se definen en la norma ANSI y por lo cual los fabricantes los implementan de forma diferente. Según el RDBMS (Relational Data Base Management System) o Sistema de Gestión de Bases de Datos relacionales que se utilice puede sufrir cambios en el manejo de los tipos de dato.

En la industria de los RDBMS manejan un estándar de tipos de datos, esto con el fin de poder convivir entre sí.

Es muy común que en las empresas se tengan ambientes del SQL Server, Oracle y MySQL, esto por decir nombres de tres RDBMS populares.

Los tipos de datos se pueden dividir ampliamente en las siguientes categorías

- Tipos de datos numéricos como int, tinyint, bigint, float, real, etc.
- Tipos de datos de fecha y hora como fecha, hora, fecha y hora, etc.
- Tipos de datos de caracteres y cadenas como char, varchar, text, etc.
- Tipos de datos de cadenas de caracteres Unicode, por ejemplo, nchar, nvarchar, ntext, etc.
- Tipos de datos binarios como binary, varbinary, etc.
- Tipos de datos varios: clob, blob, xml, cursor, tabla, etc.



### Puntos a considerar

- No todos los proveedores de bases de datos relacionales admiten todos los tipos de datos. Por ejemplo, la base de datos Oracle no admite DATETIME y MySQL no admite el tipo de datos CLOB. Por lo tanto, al diseñar el esquema de la base de datos y escribir consultas SQL, asegúrese de verificar si los tipos de datos son compatibles o no.
- Los tipos de datos que se enumeran aquí no incluyen todos los tipos de datos, estos son los tipos de datos más utilizados. Algunos proveedores de bases de datos relacionales tienen sus propios tipos de datos que pueden no aparecer aquí. Por ejemplo, Microsoft SQL Server tiene tipos de datos *money* y *smallmoney*, pero otros RDBMS no los contiene.
- Cada proveedor de bases de datos relacionales tiene su propio límite de tamaño máximo para diferentes tipos de datos, no es necesario que recuerde el límite. La idea es tener el conocimiento de qué tipo de datos se utilizarán en un escenario específico.



## Tipo de dato Carácter y Cadena de Caracteres

Datatype	Description
CHAR	Longitud fija con una longitud máxima de 8.000 caracteres
VARCHAR	Almacenamiento de longitud variable con una longitud máxima de 8.000 caracteres
VARCHAR(max)	Almacenamiento de longitud variable con caracteres máximos proporcionados, no compatible con MySQL
TEXT	Almacenamiento de longitud variable con un tamaño máximo de datos de 2 GB

### Cadena de longitud fija

Three Strings are :

- PYTHON
- HTML
- SCHEMA.ORG

P	Y	T	H	O	N	blank	blank	blank	blank	blank	blank
H	T	M	L	blank	blank	blank	blank	blank	blank	blank	blank
S	C	H	E	M	A	.	O	R	G	blank	blank

© w3resource.com

### Cadena de longitud variable

Three Strings are :

- PYTHON
- HTML
- LINUX SERVER

P	Y	T	H	O	N						
H	T	M	L								
L	I	N	U	X	blank	S	E	R	V	E	R

© w3resource.com

Ejemplo: Una tabla con columnas de cadenas de tamaño fijo y longitud variable y una cadena CLOB

```

1 CREATE TABLE test (
2   id    DECIMAL PRIMARY KEY,
3   col1 CHAR(8),      -- exactly 8 characters
4   col2 VARCHAR(100), -- up to 100 characters
5   col3 CLOB          -- very large strings
6 );

```

RDBMS Tipos de cadenas de caracteres

DBMS and version	Types
MySQL 5.7	CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM, and SET
PostgreSQL 9.5.3	CHARACTER VARYING(n), VARCHAR(n), CHARACTER(n), CHAR(n), TEXT
SQL Server	CHAR, VARCHAR, TEXT, NCHAR, NVARCHAR, NTEXT
Oracle 11g	CHAR, VARCHAR2, VARCHAR, NCHAR, NVARCHAR2, CLOB, NCLOB, LONG

## Tipo de datos Booleanos

El tipo de datos booleano comprende los distintos valores de verdad verdadero y falso. El tipo de datos booleano también admite el valor de verdad desconocido como valor nulo a menos que lo prohíba una restricción NOT NULL.

Tipo de datos	Descripción
BOOLEAN	Almacena valores de verdad, ya sea VERDADERO o FALSO.

Ejemplo: una tabla que usa un tipo booleano

```

1 CREATE TABLE test (
2   id   DECIMAL PRIMARY KEY,
3   col1 BOOLEAN
4 );

```

## Tipos Booleanos RDBMS

DBMS y versión	Types
MySQL 5.7	TINYINT(1)
PostgreSQL 9.5	BOOLEAN
SQL Server 2014	BIT
Oracle 11g	BOOLEAN

## Tipos de datos Numéricos

Los tipos de datos numéricos representan valores numéricos.

<b>Data Type</b>	<b>Descripción</b>
INTEGER	Representa un número entero. Los valores mínimos y máximos dependen del DBMS.
SMALLINT	Igual que el tipo INTEGER, excepto que puede contener un rango de valores más pequeño, dependiendo del DBMS.
BIGINT	Igual que el tipo INTEGER, excepto que puede contener un rango mayor de valores, dependiendo del DBMS.
DECIMAL(p, s)	Numérico exacto, precisión p, escala s. Un número decimal, es decir, un número que puede tener un punto decimal. El argumento de tamaño tiene dos partes: precisión y escala. La escala no puede exceder la precisión. La precisión es lo primero y una coma debe separarse del argumento de escala.
NUMERIC(p, s)	Numérico exacto, precisión p, escala s. La precisión máxima depende del DBMS.
FLOAT(p)	Precisión de mantisa numérica aproximada p. La precisión es mayor o igual a 1 y la precisión máxima depende del DBMS.
REAL	Igual que el tipo FLOAT excepto que el DBMS define la precisión.



DOUBLE PRECISION	Igual que el tipo FLOAT (DBMS define la precisión) pero mayor que el de REAL.
---------------------	---

## Precisión y Escala

Tipo de Dato	From	To
bit	0	1
tinyint	0	255
smallint	-32,768	32,767
int	-2,147,483,648	2,147,483,647
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
decimal	1E+38	10 <sup>38</sup> -1
numeric	1E+38	10 <sup>38</sup> -1
float	-1.79E + 308	1.79E + 308
real	-3.40E + 38	3.40E + 38

Ejemplo: una tabla que usa tipos de datos numéricos

```

1 CREATE TABLE test (
2   id    DECIMAL PRIMARY KEY,
3   name  VARCHAR(100),    -- up to 100 characters
4   col1  DECIMAL(5,2),    -- three digits before the decimal and two behind
5   col2  SMALLINT,       -- no decimal point
6   col3  INTEGER,        -- no decimal point
7   col4  BIGINT,         -- no decimal point.
8   col5  FLOAT(2),       -- two or more digits after the decimal place
9   col6  REAL,
10  col7  DOUBLE PRECISION
11 );
    
```

## RDBMS Tipos Numéricos

DBMS y versión	Tipo
MySQL 5.7	INTEGER(TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, INTEGER) FIXED-POINT(DECIMAL, NUMERIC) FLOATING-POINT(FLOAT, DOUBLE) BIT-VALUE(BIT),
PostgreSQL 9.5.3	SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE PRECISION, SMALLSERIAL, SERIAL, BIGSERIAL
SQL Server	EXACT NUMERICS(BIGINT, BIT, DECIMAL, INT, MONEY, NUMERIC, SMALLINT, SMALLMONEY, TINYINT) APPROXIMATE NUMERICS(FLOAT, REAL )
Oracle	NUMBER FLOATING-POINT(BINARY_FLOAT, BINARY_DOUBLE)



## Tipos de dato Fecha y Hora

Tipo de datos	Descripción
DATE	Almacena la fecha en el formato AAAA-MM-DD
TIME	Almacena la hora en el formato HH: MI: SS
DATETIME	Almacena información de fecha y hora en el formato AAAA-MM-DD HH: MI: SS
TIMESTAMP	Almacena el número de segundos transcurridos desde la época de Unix ('1970-01-01 00:00:00' UTC)
YEAR	Almacena el año en formato de 2 o 4 dígitos. Rango 1901 a 2155 en formato de 4 dígitos. Rango de 70 a 69, que representa de 1970 a 2069.

## Ejemplo de tabla usando Tipo de datos Fecha y Hora

```

1 CREATE TABLE test (
2   id   DECIMAL PRIMARY KEY,
3   col1 DATE,    -- store year, month and day (Oracle: plus hour, minute and seconds)
4   col2 TIME,
5   col3 TIMESTAMP(9), -- a timestamp with 9 digits after the decimal of seconds
6   col4 TIMESTAMP WITH TIME ZONE -- a timestamp including the name of a timezone
7 );

```

## RDBMS Tipo dato Fecha y Hora

DBMS y versión	Tipo
MySQL 5.7	DATE, TIME, DATETIME, TIMESTAMP, YEAR
PostgreSQL 9.5.3	TIMESTAMP [ WITHOUT TIME ZONE ], TIMESTAMP WITH TIME ZONE, DATE, TIME[ WITHOUT TIME ZONE ], TIME WITH TIME ZONE,
SQL Server	DATE, DATETIME2, DATETIME, DATETIMEOFFSET, SMALLDATETIME, TIME
Oracle	DATE, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE

## Intervalo de Tiempo

Almacena la cantidad de tiempo entre dos valores de fecha y hora. Por ejemplo, entre las 10:00 y las 12:30 es un intervalo de 02:30 (2 horas y 30 minutos). Hay dos clases de intervalos. El primero, llamado intervalos año-mes, no incluye campos que no sean AÑO y MES, aunque no ambos son obligatorios. La otra clase denominada intervalos de tiempo diurno que puede incluir cualquier campo que no sea AÑO o MES.

Tipo de datos	Descripción
Year-month	No incluye campos que no sean AÑO y MES, aunque no ambos son obligatorios.
Day-time	Incluye cualquier campo que no sea AÑO o MES. Estos intervalos pueden contener un valor de día, valor de hora, valor de minuto,

	valor de segundo o alguna combinación de los mismos.
--	--

Campos en valores Intervalo año – mes:

Palabra Clave	Sentido
YEAR	years
MONTH	months

Campos en valores de INTERVALO diurno:

Palabra Clave	Sentido
DAY	Dias
HOUR	Horas
MINUTE	Minutos
SECOND	Segundos y posiblemente fracciones de segundo

Valores válidos para campos en valores de INTERVALO:

Palabra clave	Valores válidos de los campos INTERVALO
YEAR	Sin restricciones, excepto por <precisión de campo líder de intervalo>
MONTH	Meses (en años) (0-11)
DAY	Sin restricciones, excepto por <precisión de campo líder de intervalo>
HOUR	Horas (en días) (0-23)

MINUTE	Minutos (en horas) (0-59)
SECOND	Segundos (en minutos) (0-59,999 ...)

Operadores válidos que involucran fechas e intervalos:

Operando	Operador	Operando
Datetime	-	Datetime
Datetime	+ or -	Interval
Interval	+	Datetime
Interval	+ or -	Interval
Interval	* or /	Numeric
Numeric	*	Interval

Ejemplo: una tabla que usa tipos de datos de intervalo

```

1 CREATE TABLE test (
2   id    DECIMAL PRIMARY KEY,
3   col1 INTERVAL YEAR TO MONTH,
4   col2 INTERVAL DAY TO SECOND(6)  -- an interval with 6 digits after the decimal of seconds
5 );
  
```

Tipos de intervalo DBMS:

DBMS y versión	Tipo
MySQL 5.7	NOT SUPPORTED
PostgreSQL 9.5	INTERVAL
SQL Server 2014	NOT SUPPORTED

Oracle 11g	INTERVAL YEAR TO MONTH INTERVAL DAY TO SECOND
------------	--

## Tipos de datos de cadenas y caracteres Unicode SQL

Según unicode.org Unicode proporciona un número único para cada carácter, sin importar la plataforma, sin importar el programa, sin importar el idioma. El estándar Unicode ha sido adoptado por líderes de la industria como Apple, HP, IBM, JustSystems, Microsoft, Oracle, SAP, Sun, Sybase, Unisys y muchos otros. Unicode es requerido por estándares modernos como XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, etc., y es la forma oficial de implementar ISO / IEC 10646. Es compatible con muchos sistemas operativos, todos los navegadores modernos y muchos otros productos. El surgimiento del estándar Unicode y la disponibilidad de herramientas que lo respaldan se encuentran entre las tendencias de tecnología de software globales recientes más importantes.

La incorporación de Unicode en sitios web y aplicaciones cliente-servidor o de varios niveles ofrece ahorros de costos significativos en comparación con el uso de conjuntos de caracteres heredados. Unicode permite que un solo producto de software o un solo sitio web se oriente a múltiples plataformas, idiomas y países sin necesidad de reingeniería. Permite que los datos se transporten a través de muchos sistemas diferentes sin dañarlos.

Tipo de datos	Descripción
NCHAR	Longitud fija con una longitud máxima de 4000 caracteres
NVARCHAR	Almacenamiento de longitud variable con una longitud máxima de 4000 caracteres

NVARCHAR (máx.)	Almacenamiento de longitud variable con caracteres máximos proporcionados
NTEXT	Almacenamiento de longitud variable con un tamaño máximo de datos de 1 GB

Tenga en cuenta que los tipos de datos anteriores no son compatibles con la base de datos MySQL.

### Tipos de datos binarios SQL

Tipo de datos	Descripción
BINARY	Longitud fija con longitud máxima de 8.000 bytes
VARBINARY	Almacenamiento de longitud variable con una longitud máxima de 8.000 bytes
VARBINARY (máx.)	Almacenamiento de longitud variable con bytes máximos proporcionados
IMAGE	Almacenamiento de longitud variable con un tamaño máximo de datos binarios de 2 GB

### Tipo de dato Cadena Binario grande

Una cadena binaria es una secuencia de octetos que no tiene un conjunto de caracteres ni una colación asociada y se describe mediante un descriptor de tipo de datos binarios.

Tipo de Dato	Descripción
BINARY LARGE OBJECT (BLOB).	BLOB stores a long sequence of bytes.

Ejemplo: una tabla con columnas de datos binarios de tamaño fijo y variable y un BLOB





```

1 CREATE TABLE test (
2   id   DECIMAL PRIMARY KEY,
3   col1 BINARY(8),      -- exactly 8 byte
4   col2 VARBINARY(140), -- up to 140 byte
5   col3 BLOB           -- very large data: store picture, sound ...
6 );
    
```

### RDBMS Tipo Binario

DBMS y Versión	Tipo
MySQL 5.7	TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
PostgreSQL 9.5	BYTEA
SQL Server 2014	BLOB
Oracle 11g	LOB

### Tipos de datos Misceláneos SQL

Tipo de datos	Descripción
CLOB	Objetos de caracteres grandes que pueden contener hasta 2 GB
XML	para almacenar datos xml
JSON	para almacenar datos JSON

## Conclusiones y recomendaciones

Esta lectura constituyen un apoyo en el proceso de aprendizaje, así mismo para apoyarse aun mas en el proceso, se debe utilizar los libros,

- Bertone, N. (2017). *Introducción a las bases de datos: fundamentos y diseño*. Pearson Educación
- Silberschatz, A. & Sudarshan, S. (2018). *Fundamentos de bases de datos* (6a. ed.). McGraw-Hill Interamericana.
- Bernal, Nieto. (2017). *Diseño de base de datos*. Universidad del Norte
- Pulido Romero, E. Escobar Domínguez, Ó. y Núñez Pérez, J. Á. (2019). *Base de datos*. Grupo Editorial Patria.

Asi mismo, todo el material investigativo por tema que el estudiante utilice para su aprendizaje es permitido, pero deben prestar atencion que los mismos no tengan mas 5 años de publicado.

## Referencias bibliográficas

- Bertone, N. (2017). *Introducción a las bases de datos: fundamentos y diseño*. Pearson Educación
- Silberschatz, A. & Sudarshan, S. (2018). *Fundamentos de bases de datos* (6a. ed.). McGraw-Hill Interamericana.
- Bernal, Nieto. (2017). *Diseño de base de datos*. Universidad del Norte
- Pulido Romero, E. Escobar Domínguez, Ó. y Núñez Pérez, J. Á. (2019). *Base de datos*. Grupo Editorial Patria.
  
- w3resource. (2020). SQL Data Types. Dirección donde se extrajo el documento: <https://www.w3resource.com/sql/data-type.php#CHAR>
- Microsoft.com. (2020). Tipos de datos de SQL Server predeterminados. Dirección donde se extrajo el documento: <https://docs.microsoft.com/es-es/sql/connect/php/default-sql-server-data-types?view=sql-server-ver15>
- Oracle.com. (2020). Tipos de Datos Soportados. Dirección donde se extrajo el documento: <https://docs.cloud.oracle.com/es-ww/iaas/data-safe/doc/supported-data-types.html>
- Blog de Ceupe (2020). LENGUAJE DE CONSULTA SQL. Dirección donde se extrajo el documento: <https://www.ceupe.com/blog/lenguaje-de-consulta-sql.html>
- Cambio Digital. (2020). ¿Qué es SQL? El primer lenguaje de analítica de datos. Dirección donde se extrajo el documento: <https://cambiodigital-ol.com/2020/02/que-es-sql-el-primer-lenguaje-de-analitica-de-datos/>



[www.usanmarcos.ac.cr](http://www.usanmarcos.ac.cr)

San José, Costa Rica