

# Sentencias SQL

---

Laboratorio Base de Datos I

# Objetivos de la Clase

---

- ☐ SQL.
- ☐ Tipos de sentencias SQL
- ☐ Describir los tipos de dato que se pueden utilizar al especificar la definición de columnas.
- ☐ Sentencias SQL Select.

# Lenguaje SQL

---

- ❑ **SQL:** Lenguaje de consulta estructurado o SQL (por sus siglas en inglés **structured query language**) es un lenguaje de programación que permite realizar diferentes operaciones sobre bases de datos relacionales.
- ❑ En la actualidad el SQL ANSI es el estándar *de facto* de la inmensa mayoría de los SGBD comerciales.
- ❑ Otro lenguaje de acceso a datos: LINQ (**Language Integrated Query**): toma la estructura y el modelo de la base de datos relacional y las convierte en un modelo de objetos a través de LINQ to SQL.
- ❑ **Transact-SQL** es el lenguaje que se utiliza para administrar instancias del **SQL Server Database Engine** (Motor de base de datos de SQL Server), para crear y administrar objetos de base de datos, y para insertar, recuperar, modificar y eliminar datos.

# Lenguaje SQL puro (ANSI)

## Clasificación

---

- ☐ DCL: Control de datos
  - Grant
  - Revoke
- ☐ DDL: Definición de datos
  - Create
  - Alter
  - Drop
- ☐ DML: Manipulación de datos
  - Select
  - Insert
  - Update
  - Delete
  - Manipulación de Transacciones:
    - ☐ BEGIN TRANSAC
    - ☐ COMMIT,ROLLBACK

# Lenguaje SQL extendido

---

- ☐ Transact SQL, PL/SQL
- ☐ Variables
  - Declare
  - Set
  - Select
- ☐ Operadores
  - Aritméticos
  - Lógicos
  - Comparativos
  - De Concatenación

# Lenguaje SQL extendido

---

- Comentarios
  - De Línea: --
  - De Bloque: /\* \*/
- Control de Flujo
  - Begin / End
  - If / Else
  - Return
  - While
  - Case

# Escritura de Sentencias SQL

---

- ☐ Las sentencias SQL no son sensibles a mayúsculas/minúsculas.
- ☐ Las sentencias SQL pueden ocupar una o más líneas.
- ☐ Las palabras claves no se pueden abreviar ni dividir entre líneas.
- ☐ Las cláusulas suelen estar colocadas en líneas separadas.
- ☐ Los sangrados se utilizan para mejorar la legibilidad.

# Tipos de Datos en SQL Server

---

- ❑ Las tablas tienen columnas, y las columnas se definen en base a un tipo de datos; los tipos de datos acotan el **tipo y tamaño** de la información que se guardará en una columna.
- ❑ La importancia de la elección de los tipos de datos reside en el almacenamiento que ocupa; para varios cientos de filas, el tamaño no es tan crucial, pero cuantas más filas se añadan a la tabla, mayor será la repercusión en el rendimiento de las operaciones de E/S.
- ❑ **Cada RDBMS maneja sus propios tipos de datos.**



# Tipos de Datos en SQL Server

---

Tipos de datos numéricos exactos que utilizan datos enteros.

Tipo de datos	Intervalo	Almacenamiento
bigint	De $-2^{63}$ (-9.223.372.036.854.775.808) a $2^{63}-1$ (9.223.372.036.854.775.807)	8 bytes
int	De $-2^{31}$ (-2.147.483.648) a $2^{31}-1$ (2.147.483.647)	4 bytes
smallint	De $-2^{15}$ (-32.768) a $2^{15}-1$ (32.767)	2 bytes
tinyint	De 0 a 255	1 byte

# Tipos de Datos en SQL Server

---

Tipos de datos numéricos que tienen precisión y escala fijas.

**decimal[ (p[ ,s] )]** y **numeric[ (p[ ,s] )]**

Cuando se utiliza la precisión máxima, los valores permitidos están comprendidos entre  $-10^{38} + 1$  y  $10^{38} - 1$ . Numeric equivale funcionalmente a decimal.

p (precisión)

El número total máximo de dígitos decimales que se puede almacenar, tanto a la izquierda como a la derecha del separador decimal.  $1 \leq p \leq 38$ . La precisión predeterminada es 18.

s (escala)

El número máximo de dígitos decimales que se puede almacenar a la derecha del separador decimal. La escala debe ser un valor comprendido entre 0 y p. Para especificar la escala es necesario haber especificado la precisión.

**Ej:** Decimal(8,2). 8 enteros y 2 decimales.

# Tipos de Datos en SQL Server

---

## Tipos de datos de caracteres

### **Char(n)** y **varchar(n)**

Los tipos de datos caracter se puede definir de longitud fija y de longitud variable.

Los de longitud fija son **char(n)** y su tamaño lo define el valor que tenga n. Por ejemplo, una columna char(15) ocupa 15 bytes.

Los de longitud variable son **varchar(n)**, y su tamaño lo define la longitud de la columna guardada; por ejemplo una columna varchar(250), que guarda el valor "columna variable" el almacenamiento que ocupa es 16 bytes.

En caso de desear valores Unicode, deberás anteponer al tipo de datos la letra n, siendo los tipos nchar, o nvarchar. La principal diferencia con los tipos de datos no-unicode, es que utilizan el doble de bytes. Por ejemplo, el texto "Tutorial", en una columna varchar(100) ocuparía 8 bytes, mientras que siendo unicode ocuparía 16 bytes.

Ej: char(20) o varchar(20). Permite guardar hasta 20 caracteres.

# Tipos de Datos en SQL Server

---

## Tipos de datos fecha

### **smalldatetime y datetime**

Son los tipos de datos utilizados para representar la fecha y la hora. El valor internamente se almacena como un valor integer, y dependiendo de la precisión utilizará 4 u 8 bytes

Tipo de datos	Intervalo	Precisión
datetime	Del 1 de enero de 1753 hasta el 31 de diciembre de 9999	3,33 milisegundos
smalldatetime	Del 1 de enero de 1900 hasta el 6 de junio de 2079	1 minuto

# Tipos de Datos en SQL Server

---

## Valores NULL

- ❑ **Indica que el valor es desconocido.**
- ❑ No hay dos valores NULL que sean iguales.
- ❑ La comparación entre dos valores NULL, o entre un valor NULL y cualquier otro valor, tiene un resultado desconocido porque el valor de cada NULL es desconocido.
- ❑ Cuando se ven los resultados de la consulta en el Editor de código de SQL Server Management Studio, los valores NULL se muestran como **(null)** en el conjunto de resultados.

# Tipos de Datos en SQL Server

---

## Otros Tipos de datos

**tipos definidos de usuario:** suele ayudar para unificar el diseño de las tablas; por ejemplo, se puede crear un tipo llamado NIF que corresponde al tipo de datos CHAR(20), y admite valores nulos.

```
CREATE TYPE NIF FROM char(20) NULL
```

## XML

A partir de la versión 2005 de SQL Server incorpora el tipo de datos nativo XML. El tipo de datos obliga a que el dato sea por lo menos bien formado (well-formed). Adicionalmente, la columna puede asociarse a un esquema XSD.

## Tipos de datos definidos de usuario en .NET

La integración del CLR, permite la posibilidad de definir tipos de datos con cualquier lenguaje .NET.

# Lenguaje de consulta SQL

---

## Proyección

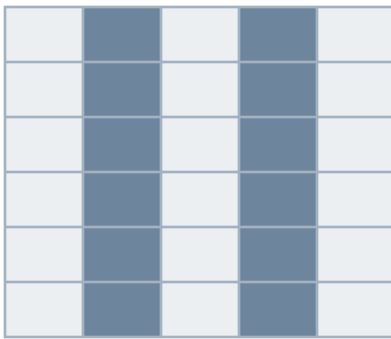



Tabla 1

## Selección

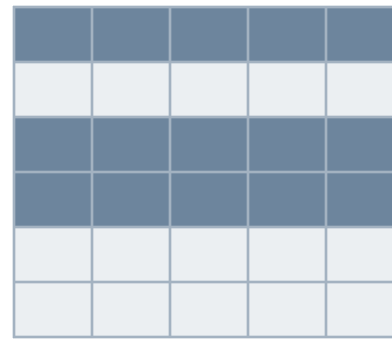



Tabla 1

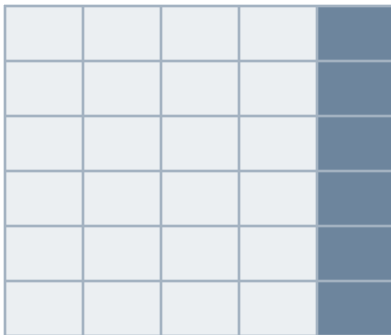



Tabla 1

## Combinación

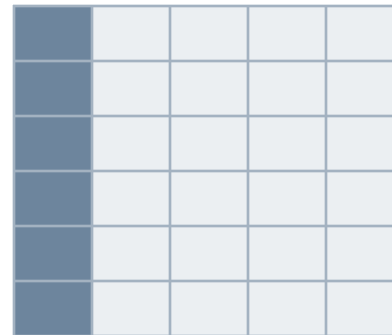



Tabla 2

# Lenguaje SQL

---

## □ SELECT

### **SELECT**

[ TOP expresión [ PERCENT ] [ WITH TIES ] ]  
<lista seleccionada> [ INTO nueva\_tabla ]  
[ **FROM** tabla ]  
[ WHERE condición ]  
[ GROUP BY expresión ]  
[ HAVING condición ]  
[ ORDER BY expresión [ ASC | DESC ] ]

## □ Join

- CROSS
- INNER
- OUTER (LEFT, RIGHT, FULL)



# Sentencias `SELECT` Básicas

## 1ra. Parte

---

```
SELECT    *|{[DISTINCT] column|expression [alias],...}  
FROM table;
```

- ❑ **SELECT** identifica las columnas
- ❑ **FROM** identifica la tabla

# Selección de Todas las Columnas

```
USE AdventureWorks;  
GO  
SELECT *  
FROM HumanResources.Department;
```

	DepartmentID	Name	GroupName	ModifiedDate
1	1	Engineering	Research and Development	1998-06-01 00:00:00.000
2	2	Tool Design	Research and Development	1998-06-01 00:00:00.000
3	3	Sales	Sales and Marketing	1998-06-01 00:00:00.000
4	4	Marketing	Sales and Marketing	1998-06-01 00:00:00.000
5	5	Purchasing	Inventory Management	1998-06-01 00:00:00.000
6	6	Research and Development	Research and Development	1998-06-01 00:00:00.000
7	7	Production	Manufacturing	1998-06-01 00:00:00.000
8	8	Production Control	Manufacturing	1998-06-01 00:00:00.000
9	9	Human Resources	Executive General and Administration	1998-06-01 00:00:00.000
10	10	Finance	Executive General and Administration	1998-06-01 00:00:00.000

(16 filas afectadas)

# Selección de Columnas Específicas

```
USE AdventureWorks;  
GO  
SELECT DepartmentID, Name  
FROM HumanResources.Department;
```

	DepartmentID	Name
1	12	Document Control
2	1	Engineering
3	16	Executive
4	14	Facilities and Maintenance
5	10	Finance
6	9	Human Resources
7	11	Information Services
8	4	Marketing
9	7	Production
10	8	Production Control
11	5	Purchasing
12	13	Quality Assurance
13	6	Research and Development
14	3	Sales
15	15	Shipping and Receiving
16	2	Tool Design

(16 filas afectadas)

# Expresiones Aritméticas

---

Pueden ser creadas con datos numéricos y operadores aritméticos.

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División

# Uso de Operadores Aritméticos

```
SELECT Name, StandardCost, ListPrice, ListPrice - StandardCost
FROM Production.Product
```

	Name	StandardC...	ListPrice	Diferencia
1	LL Mountain Seat Assembly	98,77	133,34	34,57
2	ML Mountain Seat Assembly	108,99	147,14	38,15
3	HL Mountain Seat Assembly	145,87	196,92	51,05
4	LL Road Seat Assembly	98,77	133,34	34,57
5	ML Road Seat Assembly	108,99	147,14	38,15
6	HL Road Seat Assembly	145,87	196,92	51,05
7	LL Touring Seat Assembly	98,77	133,34	34,57
8	ML Touring Seat Assembly	108,99	147,14	38,15
9	HL Touring Seat Assembly	145,87	196,92	51,05
10	HL Road Frame - Black, 58	1059,31	1431,50	372,19

# Prioridad de Operador

---



$*$   $/$   $+$   $-$

- ☐ La multiplicación y la división tienen prioridad sobre la suma y la resta.
- ☐ Los operadores de idéntica prioridad se evalúan de izquierda a derecha.
- ☐ Los paréntesis se utilizan para forzar evaluaciones prioritarias y para clarificar sentencias.

# Prioridad de Operador

```
SELECT Name, StandardCost, 5 * StandardCost + 100  
FROM Production.Product
```

	Name	StandardCost	operacion
1	LL Mountain Seat Assembly	98,77	593,85
2	ML Mountain Seat Assembly	108,99	644,95
3	HL Mountain Seat Assembly	145,87	829,35
4	LL Road Seat Assembly	98,77	593,85
5	ML Road Seat Assembly	108,99	644,95
6	HL Road Seat Assembly	145,87	829,35
7	LL Touring Seat Assembly	98,77	593,85
8	ML Touring Seat Assembly	108,99	644,95
9	HL Touring Seat Assembly	145,87	829,35
10	HL Road Frame - Black 58	1059,31	5396,55

# Uso de Paréntesis

```
SELECT Name, (5* StandardCost/100) + ListPrice  
FROM Production.Product
```

	Name	precioFinal
1	LL Mountain Seat Assembly	138,2785
2	ML Mountain Seat Assembly	152,5895
3	HL Mountain Seat Assembly	204,2135
4	LL Road Seat Assembly	138,2785
5	ML Road Seat Assembly	152,5895
6	HL Road Seat Assembly	204,2135
7	LL Touring Seat Assembly	138,2785
8	ML Touring Seat Assembly	152,5895
9	HL Touring Seat Assembly	204,2135



# Eliminación de Filas Duplicadas

---

Elimine filas duplicadas mediante la palabra clave **DISTINCT** de la cláusula **SELECT**.

```
SELECT DISTINCT ProductSubcategoryID  
FROM Production.Product;
```

# Recuperar el primer conjunto de filas de un resultado

---

## □ **TOP** (expression) [PERCENT] [ WITH TIES ]

```
SELECT TOP 10 ListPrice  
FROM Production.Product  
ORDER BY ListPrice DESC
```

(10 filas afectadas)

```
SELECT TOP 10 WITH TIES ListPrice  
FROM Production.Product  
ORDER BY ListPrice DESC
```

(13 filas afectadas)

# Contar cantidad de registros

---

□ **COUNT** ( { [ [ ALL | DISTINCT ] expression ] | \* } )

Contar las filas mediante la palabra clave COUNT(\*) de la cláusula SELECT.

```
SELECT count (*)  
FROM Production.Product;
```

```
SELECT count (DISTINCT ProductSubcategoryID)  
FROM Production.Product;
```

# Fin Tema

---

☐ ¿Preguntas?