



# GESTIÓN DE DATOS

Lenguaje de manipulación de datos (DML)  
Create, Insert, Update y Delete registros.  
Consulta Básicas con Select, From where.  
Ordenamiento de registros Order By y Top.

# Introducción

- Quiz



Universidad  
Nacional de  
Cajamarca  
*"Norte de la Universidad Peruana"*



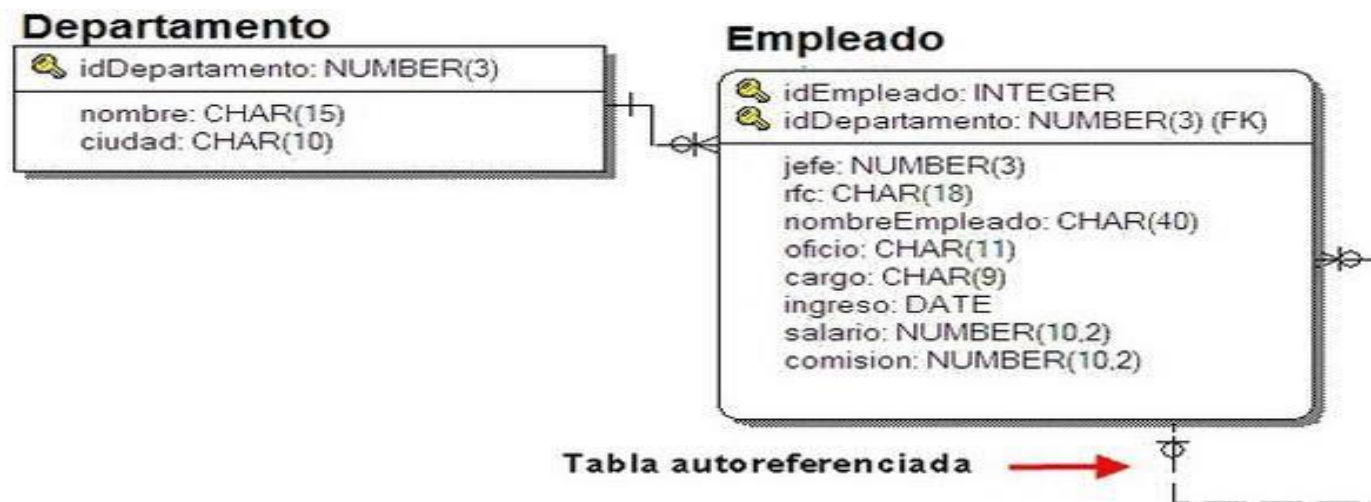


# Introducción

- Transact SQL (Structured Query Language)
- Categorías de las sentencias SQL.
- DDL Lenguaje de base de datos. y DML Lenguaje de manipulación de datos.
- Consultas condicionales.
- Operadores lógicos.
- Ejercicios prácticos.

## Lenguaje estructurado de consultas SQL

La sigla que se conoce como **SQL** corresponde a la expresión inglesa **Structured Query Language** (entendida en español como Lenguaje de Consulta Estructurado), la cual identifica a un tipo de lenguaje vinculado con la gestión de bases de datos de carácter relacional que permite la especificación de distintas clases de operaciones entre éstas. Gracias a la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla.

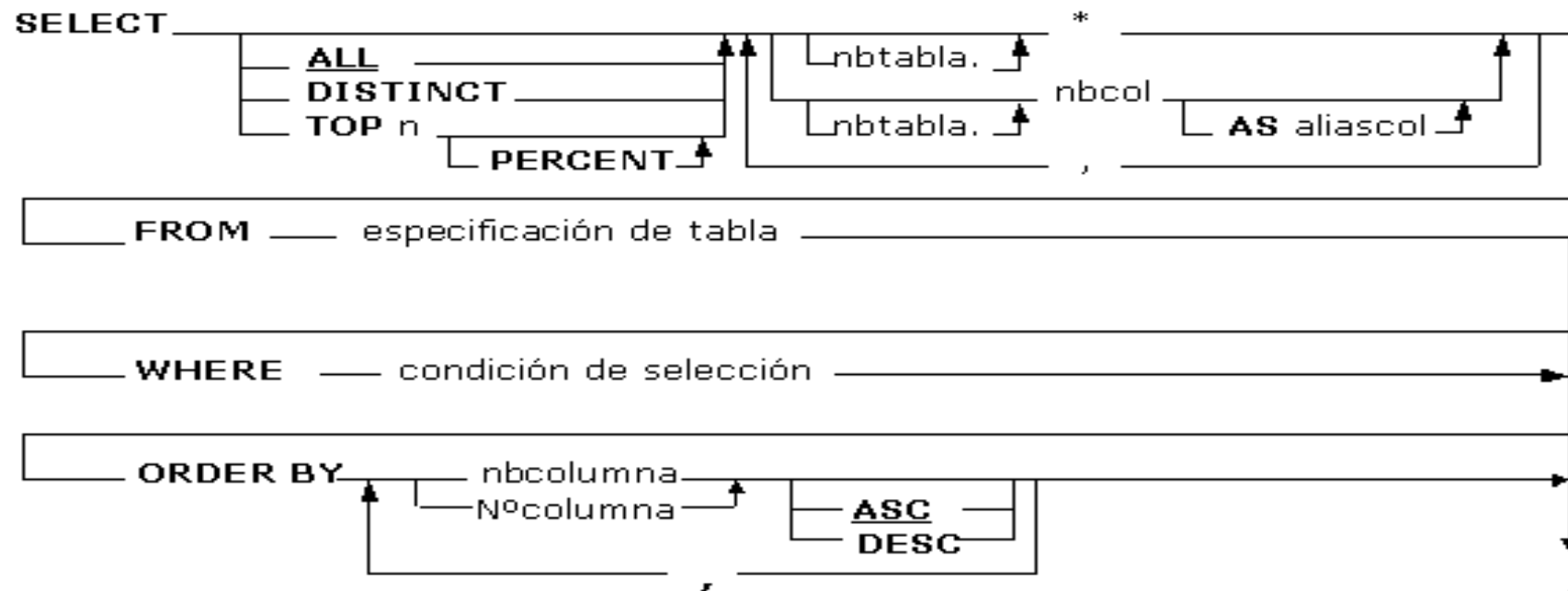




## Lenguaje de Manipulación de Datos (DML)

También es un lenguaje proporcionado por los sistemas gestores de bases de datos. En inglés, Data Manipulation Language (DML). Permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos.

La sintaxis de una sentencia de este tipo es la siguiente:



# Lenguaje Transact SQL / Data Manipulation Language



Universidad  
Nacional de  
Cajamarca  
*"Norte de la Universidad Peruana"*

## Lenguaje de Manipulación de Datos (DML)

También es un lenguaje proporcionado por los sistemas gestores de bases de datos. En inglés, Data Manipulation Language (DML). Utilizando instrucciones de SQL, permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos.

La selección sobre una tabla consiste en elegir un subconjunto de filas que cumplan (o no) algunas condiciones determinadas. La sintaxis de una sentencia de este tipo es la siguiente:

```
SELECT [*] columna1, columna2, ....  
FROM nombre-tabla  
[WHERE condición]  
[GROUP BY columna1, columna2.... ]  
[HAVING condición-selección-grupos ]  
[ORDER BY columna1 [DESC], columna2  
[DESC]... ]
```

\* indica que se obtienen todas las columnas de la/s tabla/s o vista/s indicadas en la cláusula FROM. También puede optar por colocar el nombre de la columna de la tabla.



Consultando una tabla con la sentencia Select



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

Ejm:

**SELECT \* FROM SALDOS\_AGR;**

	PERIODO	MONEDA	REGION_ID	GEO_NAME	NOM_REGION	AGENCY_ID	NOM_AGENCIA	PRODUCTO	PRODUCT_NAME	SALDO	CANTCLI	CLASE
1	201501	1	1050	COSTA	LIMA	1470	AGENCIA BARRANCA	1010	CTA.AHORRO	16668116,14	10	1
2	201501	1	1050	COSTA	LIMA	1470	AGENCIA BARRANCA	1030	CTA.CTE.	68803083,38	6	1
3	201501	1	1050	COSTA	LIMA	1470	AGENCIA BARRANCA	1050	DEP.PLAZO	1165544,18	1	3
4	201501	1	1050	COSTA	LIMA	1500	AGENCIA LA VICTORIA	1020	CTA.AHORRO	516544437,8	49	1
5	201501	2	1050	COSTA	LIMA	1500	AGENCIA LA VICTORIA	1020	CTA.AHORRO	41553594,3	11	0
6	201501	1	1050	COSTA	LIMA	1500	AGENCIA LA VICTORIA	1040	DEP.PLAZO	120183414,74	20	1
7	201501	1	1050	COSTA	LIMA	1500	AGENCIA LA VICTORIA	1050	DEP.PLAZO	363584195,28	86	1
8	201501	2	1050	COSTA	LIMA	1500	AGENCIA LA VICTORIA	1040	DEP.PLAZO	14695816,81	7	0
9	201501	2	1050	COSTA	LIMA	1500	AGENCIA LA VICTORIA	1050	DEP.PLAZO	5004741,59	1	0
10	201501	1	1050	COSTA	LIMA	1490	AGENCIA SAN DAMIAN	1050	DEP.PLAZO	5985512,37	1	2
11	201501	1	1050	COSTA	LIMA	1480	AGENCIA SUPE	1050	DEP.PLAZO	115599320,43	4	1
12	201501	1	1010	COSTA	ÁNCASH	1100	AGENCIA MORO	1050	DEP.PLAZO	861276139,06	118	1
13	201501	2	1010	COSTA	ÁNCASH	1100	AGENCIA MORO	1050	DEP.PLAZO	13136058,58	6	0
14	201501	1	1170	SIERRA	AREQUIPA	3180	AGENCIA CAYLLOMA	1050	DEP.PLAZO	196902043,74	86	1





Consultando una tabla con la sentencia Select

Ejm:

```
SELECT PERIODO, MONEDA, GEO_NAME, NOM_REGION,  
NOM_AGENCIA, SALDO FROM SALDOS_AGR;
```

	PERIODO	MONEDA	GEO_NAME	NOM_REGION	NOM_AGENCIA	SALDO
1	201501	1	COSTA	LIMA	AGENCIA BARRANCA	16668116,14
2	201501	1	COSTA	LIMA	AGENCIA BARRANCA	68803083,38
3	201501	1	COSTA	LIMA	AGENCIA BARRANCA	1165544,18
4	201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	516544437,8
5	201501	2	COSTA	LIMA	AGENCIA LA VICTORIA	41553594,3
6	201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	120183414,74
7	201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	363584195,28
8	201501	2	COSTA	LIMA	AGENCIA LA VICTORIA	14695816,81
9	201501	2	COSTA	LIMA	AGENCIA LA VICTORIA	5004741,59
10	201501	1	COSTA	LIMA	AGENCIA SAN DAMIAN	5985512,37
11	201501	1	COSTA	LIMA	AGENCIA SUPE	115599320,43
12	201501	1	COSTA	AINCASH	AGENCIA MORO	861276139,06
13	201501	2	COSTA	AINCASH	AGENCIA MORO	13136058,58
14	201501	1	SIERRA	AREQUIPA	AGENCIA CAYLLOMA	196902043,74





## Consultando una tabla / uso de alias

### Select Alias de columna

Una manera de hacer más comprensible el resultado de una consulta consiste en cambiar los encabezados de las columnas para esto se usa la expresión AS o un espacio en blanco seguido de el nombre alias de columna.

Ejm:

```
SELECT PERIODO, MONEDA, GEO_NAME AS NOMB_REGNAT, NOM_REGION AS NOMB_REGPOL,  
NOM_AGENCIA, SALDO  
FROM SALDOS_AGR;
```

O

```
SELECT PERIODO, MONEDA,  
NOM_AGENCIA, SALDO  
FROM SALDOS_AGR;
```

PERIODO	MONEDA	NOMB_REGNAT	NOMB_REGPOL	NOM_AGENCIA	SALDO
201501	1	COSTA	LIMA	AGENCIA BARRANCA	16668116,14
201501	1	COSTA	LIMA	AGENCIA BARRANCA	68803083,38
201501	1	COSTA	LIMA	AGENCIA BARRANCA	1165544,18
201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	516544437,8
201501	2	COSTA	LIMA	AGENCIA LA VICTORIA	41553594,3
201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	120183414,74
201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	363584195,28
201501	2	COSTA	LIMA	AGENCIA LA VICTORIA	14695816,81
201501	2	COSTA	LIMA	AGENCIA LA VICTORIA	5004741,59
201501	1	COSTA	LIMA	AGENCIA SAN DAMIAN	5985512,37
201501	1	COSTA	LIMA	AGENCIA SUPE	115599320,43
201501	1	COSTA	AINCASH	AGENCIA MORO	861276139,06
201501	2	COSTA	AINCASH	AGENCIA MORO	13136058,58
201501	1	SIERRA	AREQUIPA	AGENCIA CAYLLOMA	196902043,74



# Transact SQL / Operadores SQL

## Operadores Aritméticos

Los operadores aritméticos realizan operaciones matemáticas con dos expresiones de uno o más de los tipos de datos de la categoría de tipos de datos numéricos. Para obtener más información sobre las categorías de tipos de datos, vea Convenciones de sintaxis de Transact-SQL.

Operador	Significado
+ (sumar)	Suma
- (restar)	Resta
* (multiplicar)	Multiplicación
/ (dividir)	División
% (Módulo)	Devuelve el resto entero de una división. Por ejemplo, $12 \% 5 = 2$ porque el resto de 12 dividido entre 5 es 2.

# Transact SQL / Operadores SQL



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

## Operadores de asignación

El signo igual (=) es el único operador de asignación de Transact-SQL. En el siguiente ejemplo se crea la variable @MyCounter y, a continuación, el operador de asignación define @MyCounter en un valor devuelto por una expresión.

```
DECLARE @MyCounter INT;  
SET @MyCounter = 1;
```

## Operadores de comparación

Los operadores de comparación comprueban si dos expresiones son iguales. Se pueden utilizar en todas las expresiones excepto en las de los tipos de datos text, ntext o image. En la siguiente tabla se presentan los operadores de comparación Transact-SQL.

Operador	Significado
= (Igual a)	Igual a
> (Mayor que)	Mayor que
< (Menor que)	Menor que
>= (Mayor o igual que)	Mayor o igual que
<= (Menor o igual que)	Menor o igual que
<> (No igual a)	No es igual a
!= (No es igual a)	No es igual a (no es del estándar ISO)
!< (No menor que)	No es menor que (no es del estándar ISO)
!> (No mayor que)	No es mayor que (no es del estándar ISO)



# Transact SQL / Operadores SQL



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

## Operadores Lógicos

Los operadores lógicos comprueban la veracidad de alguna condición. Éstos, como los operadores de comparación, devuelven el tipo de datos Boolean con el valor TRUE, FALSE o UNKNOWN.

Operador	Significado
ALL	TRUE si el conjunto completo de comparaciones es TRUE.
AND	TRUE si ambas expresiones booleanas son TRUE.
ANY	TRUE si cualquier miembro del conjunto de comparaciones es TRUE.
BETWEEN	TRUE si el operando está dentro de un intervalo.
EXISTS	TRUE si una subconsulta contiene cualquiera de las filas.
IN	TRUE si el operando es igual a uno de la lista de expresiones.
LIKE	TRUE si el operando coincide con un patrón.
NOT	Invierte el valor de cualquier otro operador booleano.
OR	TRUE si cualquiera de las dos expresiones booleanas es TRUE.
SOME	TRUE si alguna de las comparaciones de un conjunto es TRUE.

## Operadores de Cadena

SQL Server proporciona los operadores de cadena siguientes. Los operadores de concatenación de cadenas pueden combinar dos o más cadenas o columnas de caracteres o binarias, o una combinación de cadenas y nombres de columna en una expresión. Los operadores de cadena de caracteres comodín pueden coincidir con uno o más caracteres en una operación de comparación de cadenas como LIKE y PATINDEX.

+ (Concatenación de cadenas)

+= (Concatenación de cadenas)

% (Comodín - Caracteres para coincidir)

[ ] (Comodín - Caracteres para coincidir)

[^] (Comodín - Caracteres para no coincidir)

\_ (Comodín - Un carácter para coincidir)



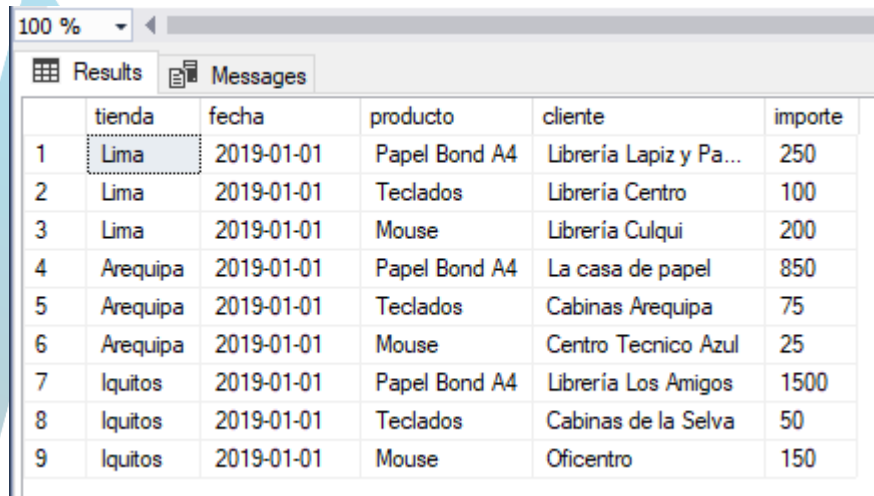
# Transact SQL / Select



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

Ahora utilizamos la sentencia SELECT para obtener los datos de la tabla.

```
SELECT * FROM ventas;
```



	tienda	fecha	producto	cliente	importe
1	Lima	2019-01-01	Papel Bond A4	Librería Lapiz y Pa...	250
2	Lima	2019-01-01	Teclados	Librería Centro	100
3	Lima	2019-01-01	Mouse	Librería Culqui	200
4	Arequipa	2019-01-01	Papel Bond A4	La casa de papel	850
5	Arequipa	2019-01-01	Teclados	Cabinas Arequipa	75
6	Arequipa	2019-01-01	Mouse	Centro Tecnico Azul	25
7	Iquitos	2019-01-01	Papel Bond A4	Librería Los Amigos	1500
8	Iquitos	2019-01-01	Teclados	Cabinas de la Selva	50
9	Iquitos	2019-01-01	Mouse	Oficentro	150

Observe:

- El carácter asterisco reemplaza a todas las columnas de la tabla
- Se muestra el numero de registro que devuelve la consulta

Sintaxis:

```
SELECT column1, column2, ...  
FROM table_name;  
WHERE condicion
```



# Transact SQL / Select



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

Puede explorar las tablas, al  
ubicar una Base de datos

Object Explorer

Connect

1

2

dbo.CustomerDemographics

dbo.Customers

dbo.Employees

Columns

- EmployeeID (PK, int, not null)
- LastName (nvarchar(20), not null)
- FirstName (nvarchar(10), not null)
- Title (nvarchar(30), null)
- TitleOfCourtesy (nvarchar(25), null)
- BirthDate (datetime, null)
- HireDate (datetime, null)
- Address (nvarchar(60), null)
- City (nvarchar(15), null)
- Region (nvarchar(15), null)
- PostalCode (nvarchar(10), null)
- Country (nvarchar(15), null)
- HomePhone (nvarchar(24), null)
- Extension (nvarchar(4), null)
- Photo (image, null)
- Notes (ntext, null)
- ReportsTo (FK, int, null)
- PhotoPath (nvarchar(255), null)

Keys

SQLQuery1.sql - DE...5N58\Usuario (52)\*

```
select * from Employees
```

Results

	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate
1	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000	1992-05-01 00:00:00.000
2	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14 00:00:00.000
3	3						
4	4						
5	5						
6	6						
7	7						
8	8						
9	9						

Puede escribir consultas  
directamente, mediante la  
opción de Nueva Consulta /  
New Query

SQLQuery1.sql - DE...5N58\Usuario (52)\*

```
select [EmployeeID], [FirstName] + ' ' + [LastName] from Employees
```

Results

	EmployeeID	(No column name)
1	1	Nancy Davolio
2	2	Andrew Fuller
3	3	Janet Leverling
4	4	Margaret Peaco...
5	5	Steven Buchanan
6	6	Michael Suyama
7	7	Robert King
8	8	Laura Callahan
9	9	Anne Dodsworth

# Transact SQL / Select



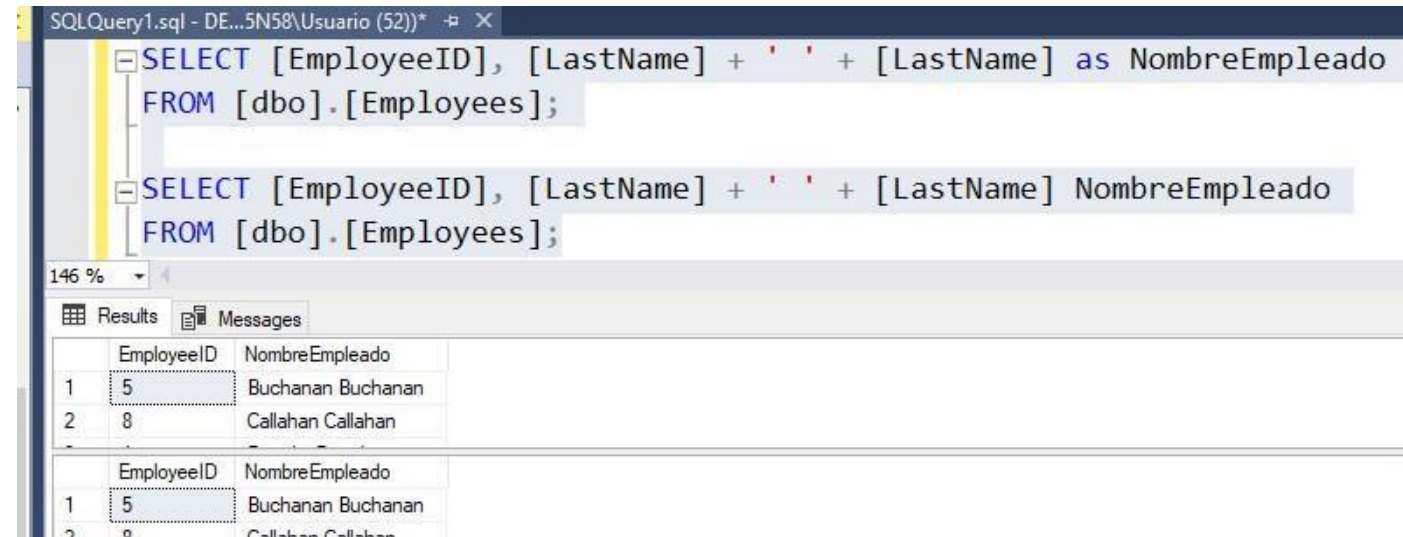
Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

## Alias de columna

Una manera de hacer más comprensible el resultado de una consulta consiste en cambiar los encabezados de las columnas para esto se usa la expresión AS o un espacio en blanco seguido de el nombre alias de columna.

```
SELECT [EmployeeID], [LastName] + ' ' + [LastName] as NombreEmpleado  
FROM [dbo].[Employees];
```

```
SELECT [EmployeeID], [LastName] + ' ' + [LastName] NombreEmpleado  
FROM [dbo].[Employees];
```



# Transact SQL / Select



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

## Columnas Calculadas

Se llama así a las columnas creadas a partir de uso de operadores y que muestran una columna con el resultado de una operación aritmética

```
SELECT [OrderID]
, [ProductID]
, [UnitPrice]
, [Quantity]
, [UnitPrice] * [Quantity]
FROM [dbo].[Order Details]
```

	OrderID	ProductID	UnitPrice	Quantity	(No column name)
1	10248	11	14,00	12	168,00
2	10248	42	9,80	10	98,00
3	10248	72	34,80	5	174,00
4	10249	14	18,60	9	167,40
5	10249	51	42,40	40	1696,00
6	10250	41	7,70	10	77,00
7	10250	51	42,40	35	1484,00
8	10250	65	16,80	15	252,00
9	10251	22	16,80	6	100,80
10	10251	57	15,60	15	234,00
11	10251	65	16,80	20	336,00



# Transact SQL / Select

## Columnas Calculadas

Se llama así a las columnas creadas a partir de uso de operadores y que muestran una columna con el resultado de una operación aritmética

```
SELECT [EmployeeID]  
, concat ([LastName], ' ', [FirstName])  
, [Title]  
, [Country]  
FROM [dbo].[Employees]
```

	EmployeeID	(No column name)	Title	Country
1	1	Davolio Nancy	Sales Representative	USA
2	2	Fuller Andrew	Vice President, Sales	USA
3	3	Leverling Janet	Sales Representative	USA
4	4	Peacock Marga...	Sales Representative	USA
5	5	Buchanan Steven	Sales Manager	UK
6	6	Suyama Michael	Sales Representative	UK
7	7	King Robert	Sales Representative	UK
8	8	Callahan Laura	Inside Sales Coordina...	USA
9	9	Dodsworth Anne	Sales Representative	UK

```
SELECT PERIODO, MONEDA, NOM_REGION_NAT NOMB_REGNAT,  
NOM_REGION_POL NOMB_REGPOL, NOM_AGENCIA,  
SALDO, CONCAT(NOM_REGION_NAT,' ',NOM_REGION_POL) AS REGIONES  
FROM SALDOS_AGR;
```

	PERIODO	MONEDA	NOMB_REGNAT	NOMB_REGPOL	NOM_AGENCIA	SALDO	REGIONES
1	201501	1	COSTA	ÁNCASH	AGENCIA MORO	861276139	COSTA ÁNCASH
2	201501	1	COSTA	LIMA	AGENCIA BARRANCA	16668116	COSTA LIMA
3	201501	1	COSTA	LIMA	AGENCIA BARRANCA	68803083	COSTA LIMA
4	201501	1	COSTA	LIMA	AGENCIA BARRANCA	1165544	COSTA LIMA
5	201501	1	COSTA	LIMA	AGENCIA SUPE	115599320	COSTA LIMA
6	201501	1	COSTA	LIMA	AGENCIA SAN DAMIAN	5985512	COSTA LIMA
7	201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	516544438	COSTA LIMA
8	201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	120183415	COSTA LIMA
9	201501	1	COSTA	LIMA	AGENCIA LA VICTORIA	363584195	COSTA LIMA
10	201501	1	SIERRA	AREQUIPA	AGENCIA CAYLLOMA	196902044	SIERRA AREQUIPA
11	201501	1	SIERRA	CAJAMARCA	AGENCIA BELLAVISTA	39084508	SIERRA CAJAMARCA



# Transact SQL / Select (Order by)

Hasta ahora, hemos visto cómo obtener datos de una tabla utilizando los comandos SELECT y WHERE. Con frecuencia, sin embargo, necesitamos enumerar el resultado en un orden particular. Esto podría ser en orden ascendente, en orden descendente, o podría basarse en valores numéricos o de texto. En tales casos, podemos utilizar la palabra clave ORDER BY para alcanzar nuestra meta.

La sintaxis para una instrucción ORDER BY es la siguiente:

```
SELECT nombre_columna  
FROM nombre_tabla  
WHERE condición  
ORDER BY nombre_columna [ASC, DESC];
```

[ ] significa que la instrucción WHERE es opcional. Sin embargo, si existe una cláusula WHERE, viene antes de la cláusula ORDER BY. ASC significa que los resultados se mostrarán en orden ascendente, y DESC significa que los resultados se mostrarán en orden descendente. Si no se especifica ninguno, la configuración predeterminada es ASC. Es posible ordenar por más de una columna. En este caso, la cláusula ORDER BY anterior se convierte en:

# Transact SQL / Where



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

Especifica la condición de filtro de las filas devueltas. Se utiliza cuando no se desea que se devuelvan todas las filas de una tabla, sino sólo las que cumplen ciertas condiciones. Lo habitual es utilizar esta cláusula en la mayoría de las consultas.

```
SELECT [EmployeeID]
,concat ([LastName], ' ', [FirstName])
, [Title]
, [Country]
FROM [dbo].[Employees]
WHERE [Country] = 'USA'
```

	EmployeeID	(No column name)	Title	Country
1	1	Davolio Nancy	Sales Representative	USA
2	2	Fuller Andrew	Vice President, Sales	USA
3	3	Leverling Janet	Sales Representative	USA
4	4	Peacock Marga...	Sales Representative	USA
5	8	Callahan Laura	Inside Sales Coordina...	USA

```
SELECT [OrderID]
, [ProductID]
, [UnitPrice]
, [Quantity]
, [UnitPrice] * [Quantity]
FROM [dbo].[Order Details]
where [ProductID] = 42
and [UnitPrice] * [Quantity] > 300
```

	OrderID	ProductID	UnitPrice	Quantity	(No column name)
1	10404	42	11,20	40	448,00
2	10463	42	11,20	50	560,00
3	10498	42	14,00	30	420,00
4	10571	42	14,00	28	392,00
5	10588	42	14,00	100	1400,00
6	10663	42	14,00	30	420,00
7	10680	42	14,00	40	560,00
8	10746	42	14,00	28	392,00





# Transact SQL / Like / Between

## LIKE:

Usada para la comparación de un modelo. Para ello utiliza los caracteres comodín especiales: "%" y "\_". Con el primero indicamos que en su lugar puede ir cualquier cadena de caracteres, y con el segundo que puede ir cualquier carácter individual (un solo carácter). Con la combinación de estos caracteres podremos obtener múltiples patrones de búsqueda. Por ejemplo:

El nombre empieza por A: Nombre LIKE 'A%'

El nombre acaba por A: Nombre LIKE '%A'

El nombre contiene la letra A: Nombre LIKE '%A%'

El nombre empieza por A y después contiene un solo carácter cualquiera: Nombre LIKE 'A\_'

El nombre empieza una A, después cualquier carácter, luego una E y al final cualquier cadena de caracteres: Nombre LIKE 'A\_E%'

**BETWEEN:** para un intervalo de valores. Por ejemplo:

Clientes entre el 30 y el 100: CodCliente BETWEEN 30 AND 100

Clientes nacidos entre 1970 y 1979: FechaNac BETWEEN '19700101' AND '19791231'

## WHERE y Operadores:

Especifica la condición de filtro de las filas devueltas. Se utiliza cuando no se desea que se devuelvan todas las filas de una tabla, sino sólo las que cumplen ciertas condiciones. Lo habitual es utilizar esta cláusula en la mayoría de las consultas.

↕ NOMB_REGNAT	↕ NOMB_REGPOL	↕ NOM_AGENCIA	↕ SALDO	↕ REGIONES
1 COSTA	LIMA	AGENCIA BARRANCA	16668116,14	COSTA LIMA
2 COSTA	LIMA	AGENCIA BARRANCA	68803083,38	COSTA LIMA
3 COSTA	LIMA	AGENCIA LA VICTORIA	516544437,8	COSTA LIMA
4 COSTA	LIMA	AGENCIA LA VICTORIA	41553594,3	COSTA LIMA
5 COSTA	LIMA	AGENCIA LA VICTORIA	120183414,74	COSTA LIMA
6 COSTA	LIMA	AGENCIA LA VICTORIA	363584195,28	COSTA LIMA
7 COSTA	LIMA	AGENCIA LA VICTORIA	14695816,81	COSTA LIMA
8 COSTA	LIMA	AGENCIA SUPE	115599320,43	COSTA LIMA

Ejm usando WHERE y Operadores:

```
SELECT NOM_REGION_NAT NOMB_REGNAT, NOM_REGION_POL NOMB_REGPOL, NOM_AGENCIA,  
SALDO, CONCAT(NOM_REGION_NAT, ' ', NOM_REGION_POL) AS REGIONES  
FROM dbo.SALDOS_AGR  
WHERE NOM_REGION_NAT = 'COSTA' AND NOM_REGION_POL = 'LIMA'  
AND SALDO >= 10000000;
```

	NOMB_REGNAT	NOMB_REGPOL	NOM_AGENCIA	SALDO	REGIONES
1	COSTA	LIMA	AGENCIA BARRANCA	16668116	COSTA LIMA
2	COSTA	LIMA	AGENCIA BARRANCA	68803083	COSTA LIMA
3	COSTA	LIMA	AGENCIA SUPE	115599320	COSTA LIMA
4	COSTA	LIMA	AGENCIA LA VICTORIA	516544438	COSTA LIMA
5	COSTA	LIMA	AGENCIA LA VICTORIA	120183415	COSTA LIMA
6	COSTA	LIMA	AGENCIA LA VICTORIA	363584195	COSTA LIMA
7	COSTA	LIMA	AGENCIA LA VICTORIA	41553594	COSTA LIMA
8	COSTA	LIMA	AGENCIA LA VICTORIA	14695817	COSTA LIMA
9	COSTA	LIMA	AGENCIA BARRANCA	11345872	COSTA LIMA
10	COSTA	LIMA	AGENCIA BARRANCA	36356441	COSTA LIMA
11	COSTA	LIMA	AGENCIA SUPE	115599320	COSTA LIMA



# Transact SQL / IN

IN( )

Para especificar una relación de valores concretos.

Por ejemplo: Ventas de los Clientes 10, 15, 30 y 75:

CodCliente **IN** (10, 15, 30, 75)

Por supuesto es posible combinar varias condiciones simples de los operadores anteriores utilizando los operadores lógicos **OR**, **AND** y **NOT**, así como el uso de paréntesis para controlar la prioridad de los operadores (como en matemáticas).





# Transact SQL / Case

- La instrucción CASE pasa por condiciones y devuelve un valor cuando se cumple la primera condición. Entonces, una vez que una condición es verdadera, dejará de leer y devolverá el resultado. Si no hay condiciones verdaderas, devuelve el valor en la cláusula ELSE.

Si no hay otra parte y no hay condiciones verdaderas, devuelve NULL.

## CASE

```
WHEN condition1 THEN result1  
WHEN condition2 THEN result2  
WHEN conditionN THEN resultN  
ELSE result
```

END;

```
SELECT OrderID, Quantity,  
CASE  
    WHEN Quantity > 30 THEN 'The quantity is greater than 30'  
    WHEN Quantity = 30 THEN 'The quantity is 30'  
    ELSE 'The quantity is under 30'  
END AS QuantityText  
FROM [dbo].[Order Details];
```

	OrderID	Quantity	QuantityText
1	10248	12	The quantity is under 30
2	10248	10	The quantity is under 30
3	10248	5	The quantity is under 30
4	10249	9	The quantity is under 30
5	10249	40	The quantity is greater than 30



**Universidad  
Nacional de  
Cajamarca**  
*"Norte de la Universidad Peruana"*

# Herramientas de Trabajo

**Online**



**SQL Server  
Management Studio**

v. 19.1





## REFERENCIAS

Insert

<https://docs.microsoft.com/es-es/sql/t-sql/statements/insert-transact-sql?view=sql-server-ver15>

Update

<https://docs.microsoft.com/es-es/sql/t-sql/queries/update-transact-sql?view=sql-server-ver15>

Where

<https://docs.microsoft.com/es-es/sql/t-sql/queries/where-transact-sql?view=sql-server-ver15>

Funciones de Agregación

<https://docs.microsoft.com/es-es/sql/t-sql/functions/aggregate-functions-transact-sql?view=sql-server-ver15>

Fin de la sesión

