



# 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

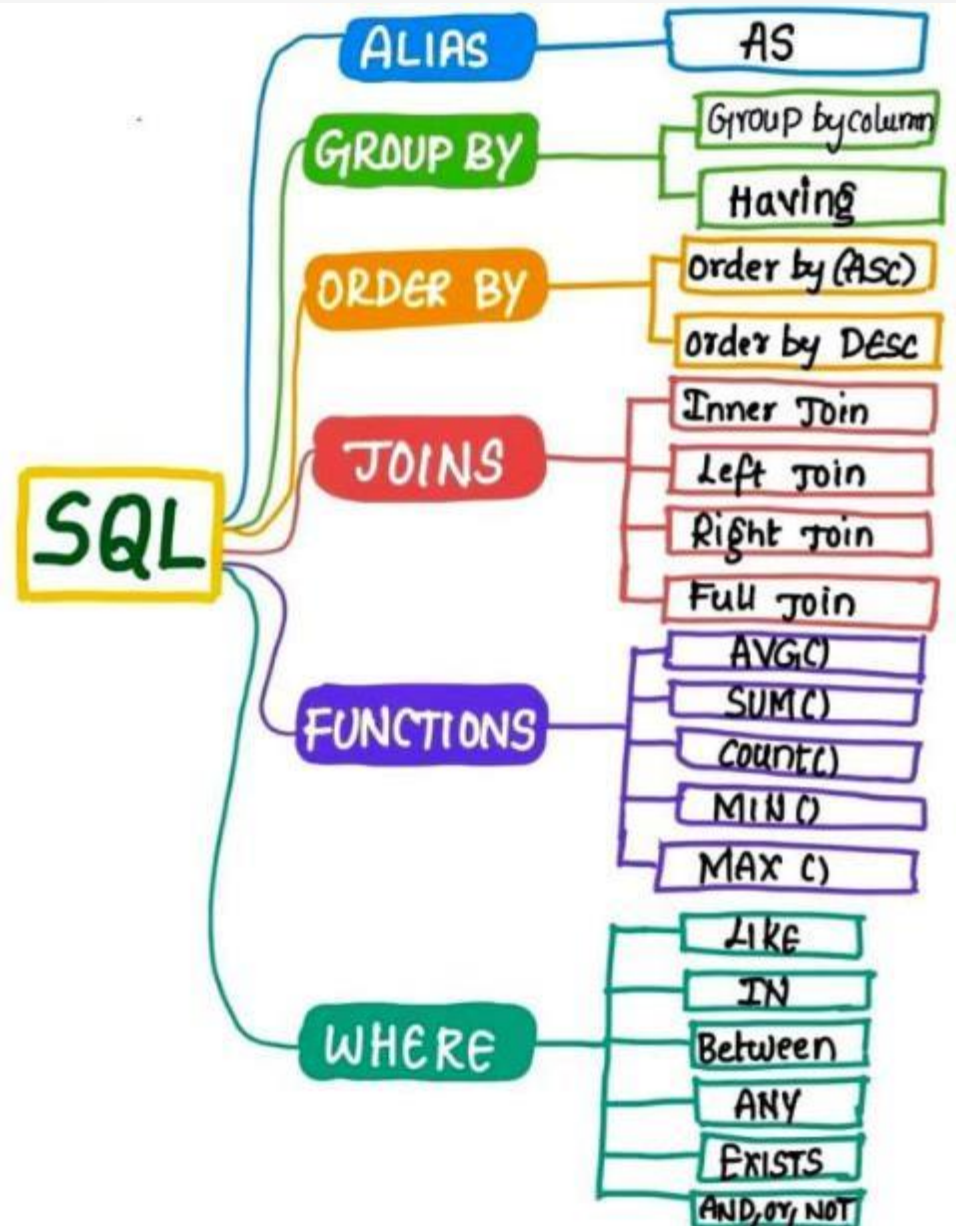
- Quiz



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



# Introducción



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"



# Renombrar AS



Universidad  
Nacional de  
Cajamarca

"Norte de la Universidad Peruana"

ALIAS — AS

Example:

An Alias is created with 'AS' Keyword.

```
SELECT column_name AS alias_name  
FROM table_name;
```

# Agrupar - Ordenar



Universidad  
Nacional de  
Cajamarca

"Norte de la Universidad Peruana"

## GROUP BY

⇒ **GROUP BY COLUMN**

Example:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

⇒ **HAVING**

Example:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

## ORDER BY

⇒ **ORDER BY (ASC)** or **ORDER BY DES**

The order by keyword is used to sort the result-set in **ascending** or **descending** order

Example:

```
SELECT column1, column2, .....
FROM table_name
ORDER BY column1, column2, ..... ASC/DES;
```

# Condiciones WHERE



Universidad  
Nacional de  
Cajamarca

"Norte de la Universidad Peruana"

## WHERE

### ⇒ LIKE

Example:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE columnN LIKE Pattern
```

### ⇒ BETWEEN

Example:

```
SELECT column_name  
FROM table_name  
WHERE column_name BETWEEN  
Value1 AND Value2;
```

### ⇒ IN

Example:

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN  
(value1, value2, ...);
```

### ⇒ ANY

Example:

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name  
Operator ANY  
(SELECT column_name  
FROM table_name
```

### ⇒ EXISTS

Example:

```
SELECT column_name(s)  
FROM table_name  
WHERE EXISTS  
SELECT column_name FROM table_name  
WHERE condition);
```

### ⇒ ALL

Example:

```
SELECT ALL column_name(s)  
FROM table_name  
WHERE condition;
```

### ⇒ AND

Example:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2 AND condition3...;
```

### ⇒ OR

Example:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 OR condition2 OR condition3...;
```

### ⇒ NOT

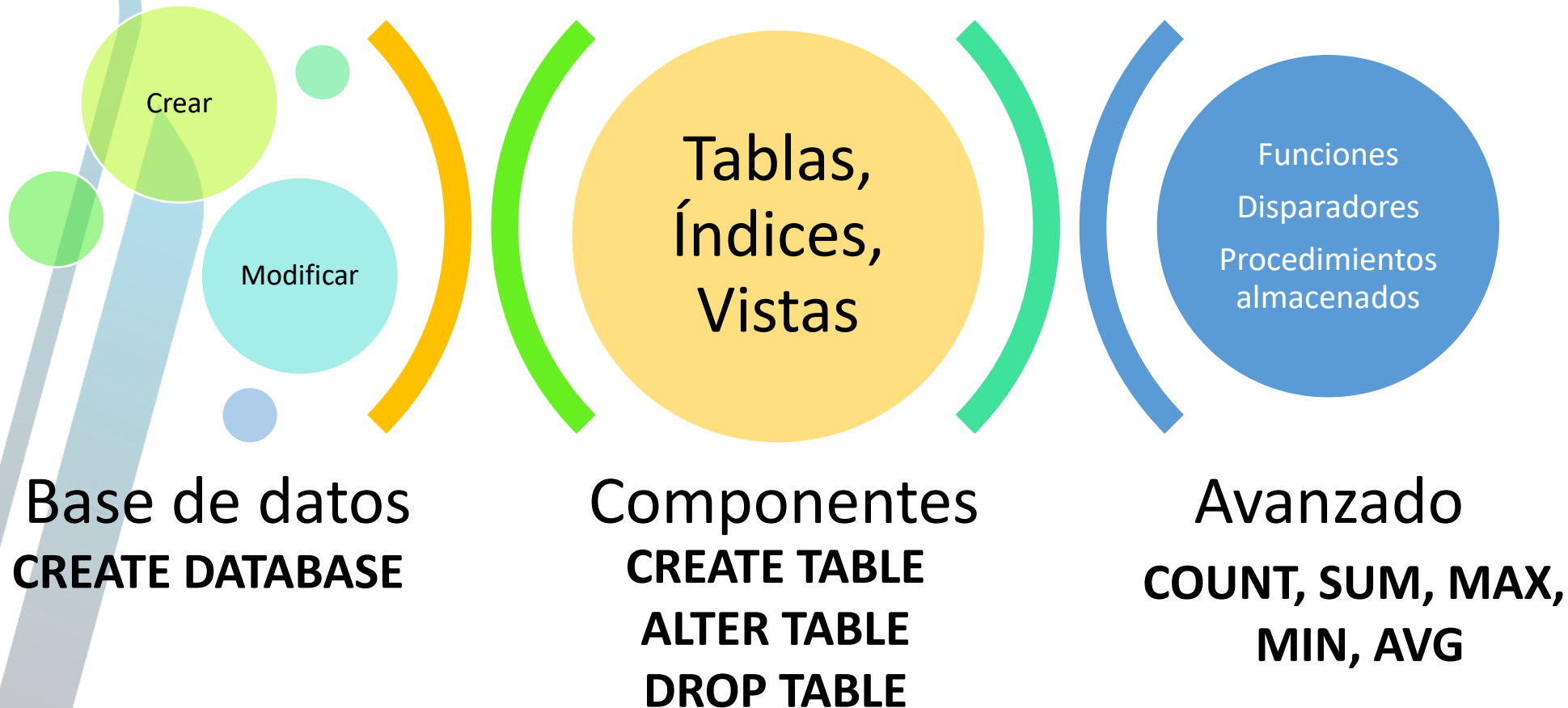
Example:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE NOT condition;
```

# ELEMENTOS, TIPOS, COMPONENTES



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"





# Elaboración de consultas

Sentencia		Origen		Condición			
<b>SELECT</b>	atributo *	<b>FROM</b>	tabla	<b>WHERE</b>	atributo	<b>ORDER BY</b>	atributo Asc Desc
	All Distinct Top						

AS  
AND, OR, \*, >, <, =, !,  
BETWEEN  
IS NULL  
IS NOT NULL  
LIKE

PRODUCTO CARTESIANO  
INNER JOIN  
LEFT JOIN / RIGHT JOIN  
YEAR, MONTH, DAY



# SUBCONSULTA, FUNCIONES FECHAS



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

Función	Descripción
GETDATE()	Devuelve la fecha y la hora actual
DATEPART()	Devuelve una sola parte de una fecha / hora
DATEADD()	Suma o resta un intervalo de tiempo especificado desde una fecha
DATEDIFF()	Devuelve el tiempo entre dos fechas
CONVERT()	Muestra datos de fecha / hora en formatos diferentes

```
SELECT NAMEPRODUCTO FROM PRODUCTO  
WHERE IDCATEGORIA = (SELECT IDCATEGORIA FROM  
CATEGORIA WHERE IDCATEGORIA=1);
```



# Tipo de Datos

## Tipo de Datos Carácter

**Char:** Los datos deben de tener una longitud fija hasta 8Kb. Ejemplo: Si queremos almacenar categorías por ejemplo A, B, C ...

Utilizaremos Char(1)

**Varchar:** Puede variar en el número de caracteres, es decir el valor asignado no es fijo, aquí SQL administra los espacios en blanco y los optimiza. Ejemplo en un Varchar(15) ocupará menos espacio el dato "Ana" que "Juan Carlos".



Microsoft®  
**SQL Server®**  
FURTHER. FORWARD. FASTER.



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"

# Tipo de Datos

## Tipo de Datos Enteros

**Int:** Números enteros desde  $-2^{31}$  (-2 147 483 648) a  $2^{31}$  (2 147 483 647).

## Tipos de Datos Fecha

**Date:** Tipo de dato que muestra la fecha en el siguiente formato 07-07-2014.







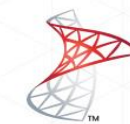
# Tipo de Datos

## Tipo de Datos Decimal

**Decimal:** Tipo de datos que se utiliza para almacenar números decimales que pueden tener hasta 38 dígitos.

## Tipo de Datos Monetario

**Money:** Cantidad monetaria positiva o negativa.



Microsoft®  
**SQL Server®**  
FURTHER. FORWARD. FASTER.

# Propiedades de Datos

**Propiedad NULL:** Hay dos formas de expresar el término NULL, al implementarlo como NULL estamos indicando que el contenido de dicha columna no es obligatorio, si se necesita especificar que el campo es obligatorio se implementará con NOT NULL.

**Propiedad IDENTITY:** Propiedad sólo aplicada a campos numéricos, ya que define un autoincremento automático de valores.



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"



Microsoft®  
**SQL Server®**  
FURTHER. FORWARD. FASTER.



# Restricciones de los Campos

**Unique:** Permite determinar que los valores registrados en una misma columna no sean idénticos, es decir se mantengan únicos. Por ejemplo el email de una persona es un campo único.

**Check:** Permite restringir el rango de valores que pueden estar permitidos ingresar en una o más columnas de una tabla.



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"



Microsoft®  
**SQL Server®**  
FURTHER. FORWARD. FASTER.





# Restricciones de los Campos

**Default:** Permite registrar un dato en un campo por defecto cuando el usuario no ingresa ningún valor, la propiedad del campo de la tabla necesariamente tiene que ser Null.

Ejemplo se puede Utilizar para los campos de tipo Date (fecha), cuando el usuario no ingresa ninguna fecha que se asigne la fecha actual.



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"



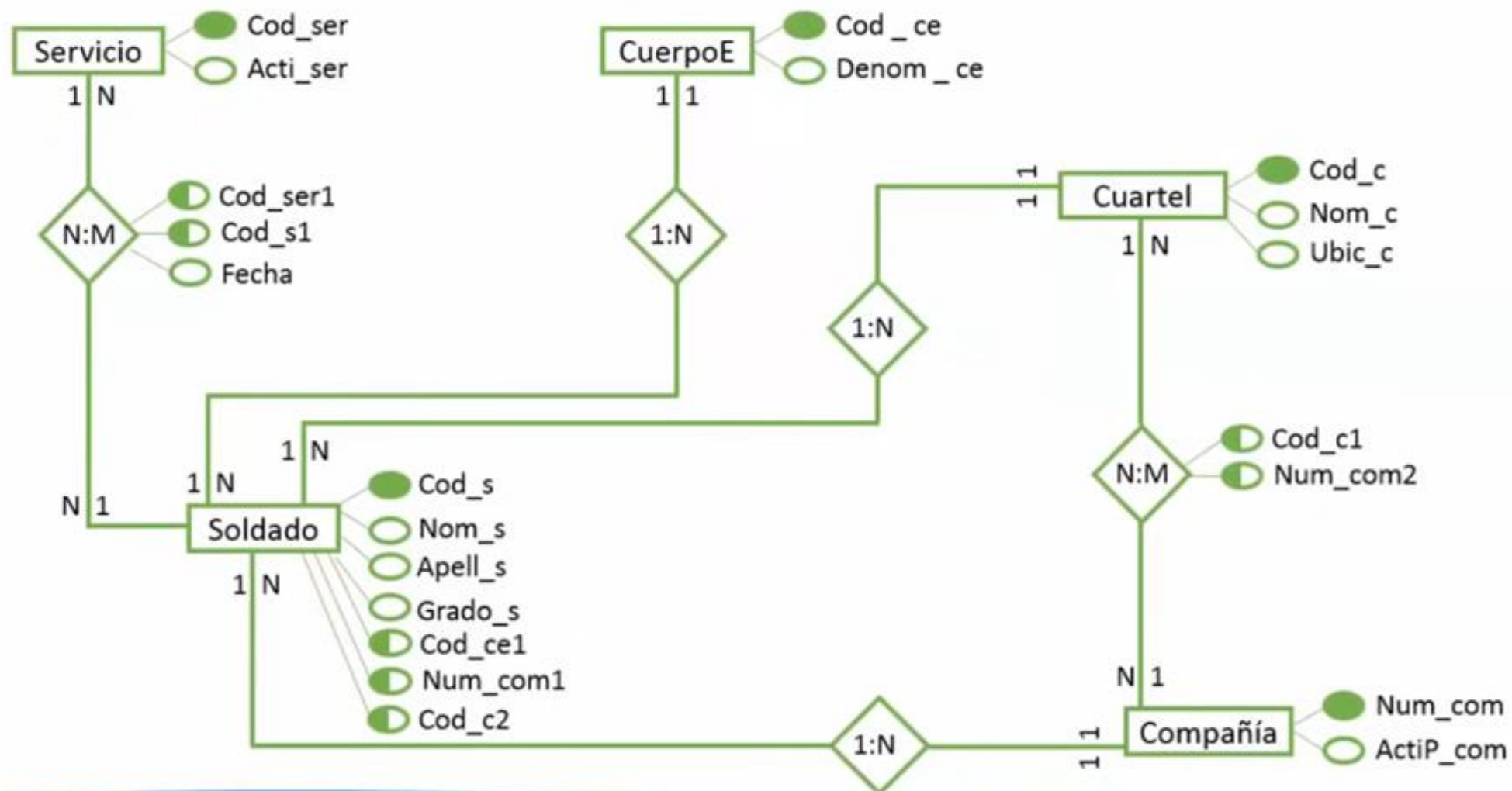
Microsoft®  
**SQL Server®**  
FURTHER. FORWARD. FASTER.



# Caso Práctico Create, Insert



Universidad  
Nacional de  
Cajamarca  
"Norte de la Universidad Peruana"





# Crear una base de datos

## Practica

Crear dos tablas, agregar relación entre las tablas, insertar datos y generar consultas CRUD

SQL  
Online

## Caso Practico

Implementar una Base de Datos con comandos DDL y DML

Microsoft  
SQL Server



Fin de la sesión

