



DATABASE FOUNDATIONS (ORACLE)

Instructor: ING. RUPAY CASTRO, E. NESTOR

SEMANA 13

Objetivo de la sesión

Programar un script para creación de una base de datos con estructura y datos.

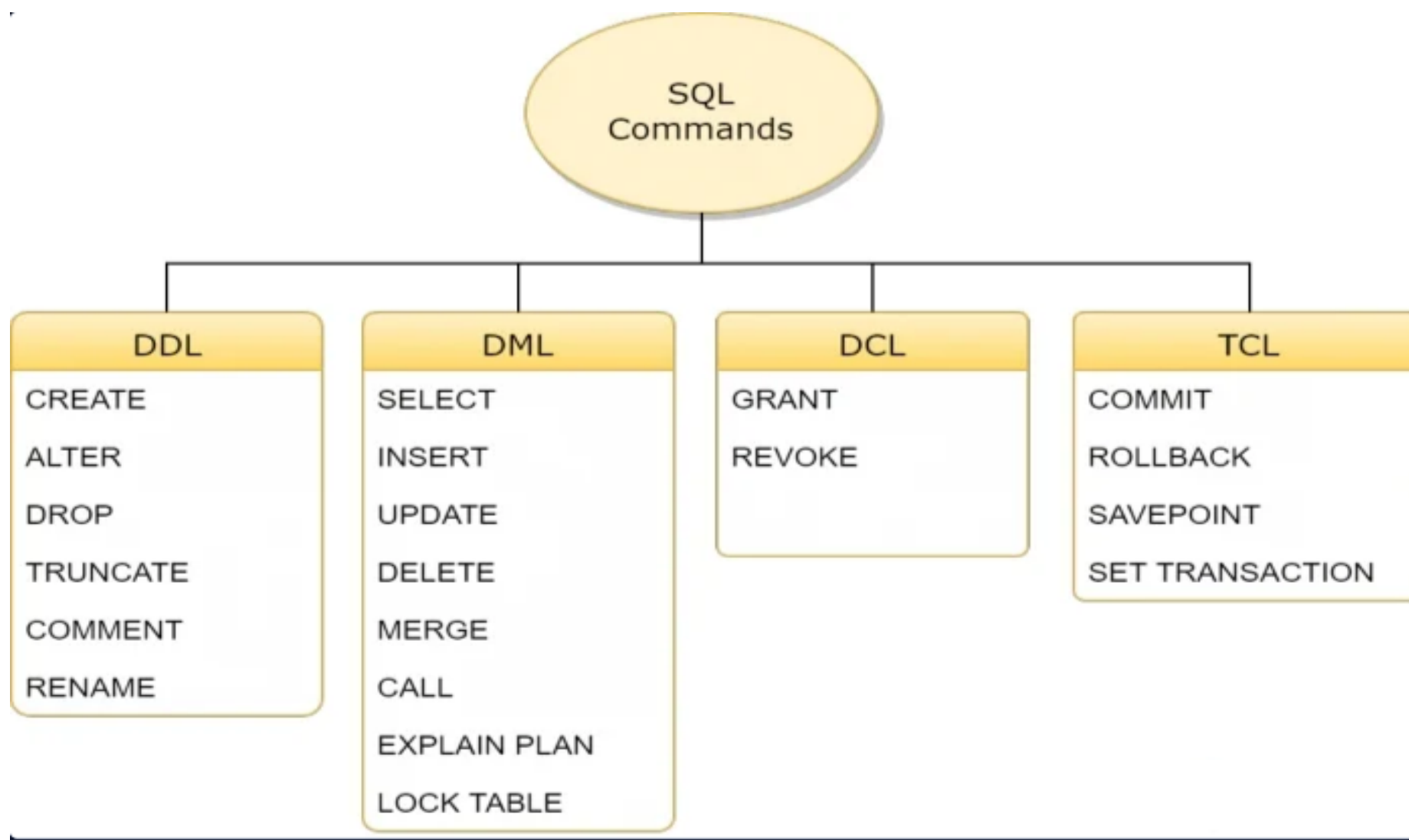


CONOCIMIENTOS PREVIOS



1. ¿Qué es un DDL?

2. ¿Qué es un DML?



DDL

(Data Definition Language o Lenguaje de Definición de Datos)

- Aquí ya se especifica el esquema de la BD, generando un diccionario de datos, las restricciones de integridad y las autorizaciones para que ciertos usuarios no vean cierto contenido.
- Sentencias DDL son las necesarias para poder modificar la BD, esquema y ESTRUCTURA de las tablas. Son las útiles para el diseño y control de objetos que se encuentran dentro de las BD.
- Existen 3 grandes comandos:
 - Create: Nos ayuda a crear bases de datos, tablas, vistas, índices, etc.
 - Alter: Ayuda a alterar o modificar entidades.
 - Drop: Nos ayuda a borrar. Hay que tener cuidado al utilizarlo.

DML

(Data Manipulation Language o Lenguaje de Manipulación de Datos)

- Lenguaje procedimental y declarativo, conjunto de instrucciones que apoyarán al proceso de construcción de la BD
- Las sentencias DML afectan los registros en una tabla. Estas son operaciones básicas que realizamos sobre datos tales como seleccionar algunos registros de una tabla, insertar nuevos registros, eliminar registros innecesarios y actualizar / modificar registros existentes.
- Opciones DML
 - SELECT: para seleccionar registros de tablas
 - INSERT: para insertar nuevos registros
 - UPDATE: para actualizar y modificar registros
 - DELETE: para eliminar registros existentes.

DCL

(Data Control Language o Lenguaje de Control de Datos)

- Las declaraciones DCL son las encargadas de controlar el acceso de los usuarios a las BD.

Opciones DDL

- GRANT: Declaración que permite a los usuarios leer / escribir en objetos que digamos de la BD.
- REVOKE: Es la que ofrece a los usuarios estar sin permiso de lectura / escritura en objetos de la BD.

TLC

(Transaction Control Language o Lenguaje de Control de Transacciones)

Instrucciones que permiten administrar transacciones y tener integridad de datos dentro de las declaraciones SQL.

Se gestiona a través de las siguientes declaraciones:

- **BEGIN Transaction:** Nos permite abrir una transacción
- **COMMIT Transaction:** Ofrece confirmar una transacción
- **ROLLBACK Transaction:** Devuelve una transacción en caso de error cometido.

OPERADORES

Prioridad	Operador	Operacion
1º	** , NOT	exponenciación, negación
2º	* , /	multiplicación, división
3º	+ , - ,	suma, resta, concatenación. Para concatenar Access se utiliza el signo +.
4º	= , != , < , > , <= , >= , IS NULL , LIKE , BETWEEN , IN	Comparación. En Access la comparación != es <>
5º	AND	conjunción
6º	OR	inclusión

FUNCIONES PREDEFINIDAS

Llamada a la función	Valor devuelto	Explicación.
ABS(-5)	5	el valor absoluto del número.
ROUND(8.66, 0)	9	el número redondeando a cero decimales
TRUC(8.66, 0). En Acces esta función no se encuentra.	8	el número truncando los decimales.
SIGN(8.88). En Acces se llama SGN()	1	ya que el número es mayor de 0
INITCAP('hola'). En Acces esta función no se encuentra.	'Hola'	cadena con el primer carácter en mayúsculas.
UPPER('hola'). En Acces se llama UCASE()	'HOLA'	cadena en mayúsculas
SUBSTR('hola', 2, 2). En Acces se llama MID()	'ol'	subcadena tomando dos posiciones desde la posición 2.
SYSDATE. En Acces se llama DATE()	01/02/00	la OPERADORES tema
USER. En Acces se llama CURRENTUSER()	CURSOSQL	el nombre del usuario en nuestro caso el que se indica.

FUNCIONES NUMÉRICA O ARIMÉTICAS

Función	Valor que devuelve.
ABS(<i>num</i>)	Valor absoluto de <i>num</i> .
MOD(<i>num1</i>, <i>num2</i>). En Access se pone: <i>num1 MOD num2</i>	Resto de la división entera.
POWER(<i>num1</i>, <i>num2</i>). En Access se pone: <i>num1 ^ num2</i> .	Devuelve <i>num1</i> elevado a <i>num2</i> .
ROUND(<i>num1</i>, <i>num2</i>)	Devuelve <i>num1</i> redondeado a <i>num2</i> decimales.
SIGN(<i>num</i>). En Access se llama SGN() .	Si <i>num</i> < 0 devuelve -1, si <i>num</i> = 0 devuelve 0, si <i>num</i> > 0 devuelve 1.
SQRT(<i>num</i>). En Access se llama SQR .	Raíz cuadrada de <i>num</i>
TRUNC(<i>num1</i>, <i>num2</i>). No se encuentra en Access	Devuelve <i>num1</i> truncado a <i>num2</i> decimales. Si se omite <i>num2</i> , a 0 decimales.

FUNCIONES DE CARACTERES

Función	Valor que devuelve.
ASCII(<i>c1</i>). En Access se llama ASC .	Código ASCII del carácter <i>c1</i> .
CHR(<i>num</i>)	Carácter correspondiente al código indicado por <i>num</i> del juego de caracteres utilizado.
CONCAT(<i>c1</i>,<i>c2</i>). Para concatenar en Access utilizamos + o &.	Concatena <i>c1</i> con <i>c2</i> . Es equivalente al operador . En Access. Pondremos <i>c1</i> & <i>c2</i> o también <i>c1</i> + <i>c2</i> .
INITCAP(<i>c1</i>). No se encuentra en Access	Devuelve <i>c1</i> poniendo en mayúscula la primera letra de cada palabra de la cadena y el resto en minúsculas.

OPERADORES en Access

LENGTH(<i>c1</i>) En Access se llama LEN() .	Longitud de <i>c1</i> .
LOWER(<i>c1</i>) En Access se llama LCASE .	La cadena <i>c1</i> en minúsculas.
UPPER(<i>c1</i>) En Access se llama UCASE .	La cadena <i>c1</i> en mayúsculas.
LPAD(<i>c1</i>, <i>n</i>, <i>c2</i>) No se encuentra en Access	<i>c1</i> se visualiza con longitud <i>n</i> y justificado a la derecha. Si <i>c1</i> < <i>n</i> , <i>c2</i> es la cadena con la que se rellena por la izda.
RPAD(<i>c1</i>, <i>n</i>, <i>c2</i>) No se encuentra en Access	Igual que LPAD pero por la derecha.
LTRIM(<i>c1</i>)	Suprime blancos a la izquierda de <i>c1</i> .
RTRIM(<i>c1</i>)	Suprime blancos a la derecha de <i>c1</i> .
SUBSTR(<i>c1</i>, <i>n</i>, <i>m</i>) En Access se llama MID()	Devuelve una subcadena a partir de <i>c1</i> comenzando en la posición <i>n</i> tomando <i>m</i> caracteres.
TRANSLATE(<i>c1</i>, <i>c2</i>, <i>c3</i>) No se encuentra en Access	Devuelve la cadena <i>c1</i> con cada ocurrencia de <i>c2</i> que sea reemplazada por <i>c3</i> .

OPERADORES en Access

FUNCIONES DE FECHA

Función	Valor que devuelve.
ADD_MONTHS(<i>f</i>, <i>n</i>) En Access se utiliza: DateAdd("y",<i>n</i>, <i>f</i>). Suma <i>n</i> años DateAdd("m",<i>n</i>,<i>f</i>). Suma <i>n</i> meses DateAdd("d",<i>n</i>,<i>f</i>). Suma <i>n</i> días	Incrementa <i>n</i> meses a la fecha <i>f</i> .. En el caso de Access podemos sumar años, días o meses a una fecha dada.
LAST_DAY(<i>f</i>) No se encuentra en Access	Último día del mes de la fecha.
MONTHS_BETWEEN (<i>f1</i>, <i>f2</i>) En Access se utiliza: DateDiff("m",<i>f1</i>,<i>f2</i>). Los meses. DateDiff("d",<i>f1</i>,<i>f2</i>). Los días. DateDiff("yyyy",<i>f1</i>,<i>f2</i>). Los años	Número de meses entre las dos fechas. El resultado puede contener decimales correspondientes a fracciones de mes. En el caso de Access podemos saber el número de días, de años y de meses.
SYSDATE. En Access se llama DATE().	Fecha actual. También la hora si se especifica en el formato. En Access es NOW() la función que visualiza la hora y la fecha.

FUNCIONES DE CONVERSIÓN

Función	Valor que devuelve.																																								
TO_NUMBER(c1) En Access se llama VAL().	Convierte una cadena a tipo numérico.																																								
TO_CHAR(n, formato)	<p>Devuelve un número en formato char según las siguientes especificaciones de formato:</p> <table><tr><th>Cod.</th><th>Ejemplo</th><th>Descripción</th></tr><tr><td>9</td><td>999</td><td>Cada 9 indica un dígito</td></tr><tr><td>0</td><td>099</td><td>Visualiza ceros a la izquierda</td></tr><tr><td>\$</td><td>\$99</td><td>Antepone el símbolo de \$</td></tr><tr><td>MI</td><td>99MI</td><td>Visualiza un '-' después de un número negativo</td></tr><tr><td>B</td><td>B99</td><td>En lugar de 0 visualiza blancos</td></tr><tr><td>,</td><td>999,999</td><td>Visualiza el separador de los miles</td></tr><tr><td>.</td><td>999,999.99</td><td>Visualiza el punto de separación de decimales</td></tr></table>	Cod.	Ejemplo	Descripción	9	999	Cada 9 indica un dígito	0	099	Visualiza ceros a la izquierda	\$	\$99	Antepone el símbolo de \$	MI	99MI	Visualiza un '-' después de un número negativo	B	B99	En lugar de 0 visualiza blancos	,	999,999	Visualiza el separador de los miles	.	999,999.99	Visualiza el punto de separación de decimales																
Cod.	Ejemplo	Descripción																																							
9	999	Cada 9 indica un dígito																																							
0	099	Visualiza ceros a la izquierda																																							
\$	\$99	Antepone el símbolo de \$																																							
MI	99MI	Visualiza un '-' después de un número negativo																																							
B	B99	En lugar de 0 visualiza blancos																																							
,	999,999	Visualiza el separador de los miles																																							
.	999,999.99	Visualiza el punto de separación de decimales																																							
TO_CHAR(fecha, formato)	<p>Devuelve cadena a partir de una fecha según las siguientes especificaciones de formato:</p> <table><tr><th>Formato</th><th>Descripción</th></tr><tr><td>YYYY</td><td>Cuatro dígitos en el año.</td></tr><tr><td>YY</td><td>Últimos dos dígitos del año (también con tres y uno)</td></tr><tr><td>YEAR</td><td>Año deletreado.</td></tr><tr><td>MM</td><td>Mes en número.</td></tr><tr><td>MONTH</td><td>Mes en letra.</td></tr><tr><td>MON</td><td>Abreviatura tres letras del mes</td></tr><tr><td>DD</td><td>Día del mes</td></tr><tr><td>DDD</td><td>Día del año</td></tr><tr><td>D</td><td>Día de la semana</td></tr><tr><td>DAY</td><td>Nombre del día de la semana</td></tr><tr><td>DY</td><td>Nombre del día con tres letras</td></tr><tr><td>WW</td><td>Semana del año</td></tr><tr><td>W</td><td>Semana del mes</td></tr><tr><td>Q</td><td>Trimestre del año</td></tr><tr><td>AM/PM</td><td>Formato 12 horas indicando AM o PM</td></tr><tr><td>HH24</td><td>Formato 24 horas</td></tr><tr><td>MI</td><td>Minutos</td></tr><tr><td>SS</td><td>Segundos</td></tr><tr><td>SSSS</td><td>Hora en segundos desde las 0 horas del día.</td></tr></table>	Formato	Descripción	YYYY	Cuatro dígitos en el año.	YY	Últimos dos dígitos del año (también con tres y uno)	YEAR	Año deletreado.	MM	Mes en número.	MONTH	Mes en letra.	MON	Abreviatura tres letras del mes	DD	Día del mes	DDD	Día del año	D	Día de la semana	DAY	Nombre del día de la semana	DY	Nombre del día con tres letras	WW	Semana del año	W	Semana del mes	Q	Trimestre del año	AM/PM	Formato 12 horas indicando AM o PM	HH24	Formato 24 horas	MI	Minutos	SS	Segundos	SSSS	Hora en segundos desde las 0 horas del día.
Formato	Descripción																																								
YYYY	Cuatro dígitos en el año.																																								
YY	Últimos dos dígitos del año (también con tres y uno)																																								
YEAR	Año deletreado.																																								
MM	Mes en número.																																								
MONTH	Mes en letra.																																								
MON	Abreviatura tres letras del mes																																								
DD	Día del mes																																								
DDD	Día del año																																								
D	Día de la semana																																								
DAY	Nombre del día de la semana																																								
DY	Nombre del día con tres letras																																								
WW	Semana del año																																								
W	Semana del mes																																								
Q	Trimestre del año																																								
AM/PM	Formato 12 horas indicando AM o PM																																								
HH24	Formato 24 horas																																								
MI	Minutos																																								
SS	Segundos																																								
SSSS	Hora en segundos desde las 0 horas del día.																																								

CLÁUSULAS

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular

Cláusula	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

FUNCIONES DE AGREGADOS

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

CONSULTAS DE PREDICADO

El predicado se incluye entre la cláusula y el primer nombre del campo a recuperar, los posibles predicados son:

Predicado	Descripción
ALL	Devuelve todos los campos de la tabla
TOP	Devuelve un determinado número de registros de la tabla
DISTINCT	Omite los registros cuyos campos seleccionados coincidan totalmente
DISTINCTROW	Omite los registros duplicados basandose en la totalidad del registro y no sólo en los campos seleccionados.

TIPO DE DATOS

BINARY	1 byte	Para consultas sobre tabla adjunta de productos de bases de datos que definen un tipo de datos Binario.
BIT	1 byte	Valores Si/No ó True/False
BYTE	1 byte	Un valor entero entre 0 y 255.
COUNTER	4 bytes	Un número incrementado automáticamente (de tipo Long)
CURRENCY	8 bytes	Un entero escalable entre 922.337.203.685.477,5808 y 922.337.203.685.477,5807.
DATETIME	8 bytes	Un valor de fecha u hora entre los años 100 y 9999.
SINGLE	4 bytes	Un valor en punto flotante de precisión simple con un rango de -3.402823×10^{38} a $-1.401298 \times 10^{-45}$ para valores negativos, 1.401298×10^{-45} a 3.402823×10^{38} para valores positivos, y 0.

TIPO DE DATOS

DOUBLE	8 bytes	Un valor en punto flotante de doble precisión con un rango de $-1.79769313486232 \times 10^{308}$ a $-4.94065645841247 \times 10^{-324}$ para valores negativos, $4.94065645841247 \times 10^{-324}$ a $1.79769313486232 \times 10^{308}$ para valores positivos, y 0.
SHORT	2 bytes	Un entero corto entre -32,768 y 32,767.
LONG	4 bytes	Un entero largo entre -2,147,483,648 y 2,147,483,647.
LONGTEXT	1 byte por carácter	De cero a un máximo de 1.2 gigabytes.
LONGBINARY	Según se necesite	De cero 1 gigabyte. Utilizado para objetos OLE.
TEXT	1 byte por caracter	De cero a 255 caracteres.

SUBCONSULTAS

Una subconsulta es una instrucción SELECT anidada dentro de una instrucción SELECT, SELECT...INTO, INSERT...INTO, DELETE, o UPDATE o dentro de otra subconsulta.

Puede utilizar tres formas de sintaxis para crear una subconsulta: comparación [ANY | ALL | SOME] (instrucción sql) expresión [NOT] IN (instrucción sql) [NOT] EXISTS (instrucción sql)

En donde:

comparación

Es una expresión y un operador de comparación que compara la expresión con el resultado de la subconsulta.

expresión

Es una expresión por la que se busca el conjunto resultante de la subconsulta.

instrucción

sql Es una instrucción SELECT, que sigue el mismo formato y reglas que cualquier otra instrucción SELECT. Debe ir entre paréntesis.

SUBCONSULTAS

Se puede utilizar una subconsulta en lugar de una expresión en la lista de campos de una instrucción SELECT o en una cláusula WHERE o HAVING. En una subconsulta, se utiliza una instrucción SELECT para proporcionar un conjunto de uno o más valores especificados para evaluar en la expresión de la cláusula WHERE o HAVING.

Se puede utilizar el predicado ANY o SOME, los cuales son sinónimos, para recuperar registros de la consulta principal, que satisfagan la comparación con cualquier otro registro recuperado en la subconsulta. El ejemplo siguiente devuelve todos los productos cuyo precio unitario es mayor que el de cualquier producto vendido con un descuento igual o mayor al 25 por ciento:

```
SELECT * FROM Productos WHERE PrecioUnidad > ANY  
(SELECT PrecioUnidad FROM DetallePedido WHERE Descuento >= 0.25);
```

CONSULTA DE UNIÓN INTERNAS

Las vinculaciones entre tablas se realizan mediante la cláusula INNER que combina registros de dos tablas siempre que haya concordancia de valores en un campo común. Su sintaxis es:

```
SELECT campos FROM tb1 INNER JOIN tb2 ON tb1.campo1 comp  
tb2.campo2
```

En donde:

tb1, tb2

Son los nombres de las tablas desde las que se combinan los registros.

campo1, campo2

Son los nombres de los campos que se combinan. Si no son numéricos, los campos deben ser del mismo tipo de datos y contener el mismo tipo de datos, pero no tienen que tener el mismo nombre.

comp

Es cualquier operador de comparación relacional : =, , <=, >=, o <>.

CONSULTAS DE COMBINACIÓN ENTRE TABLAS

Se puede utilizar una operación INNER JOIN en cualquier cláusula FROM. Esto crea una combinación por equivalencia, conocida también como unión interna. Las combinaciones Equi son las más comunes; éstas combinan los registros de dos tablas siempre que haya concordancia de valores en un campo común a ambas tablas. Se puede utilizar INNER JOIN con las tablas Departamentos y Empleados para seleccionar todos los empleados de cada departamento. Por el contrario, para seleccionar todos los departamentos (incluso si alguno de ellos no tiene ningún empleado asignado) se emplea LEFT JOIN o todos los empleados (incluso si alguno no está asignado a ningún departamento), en este caso RIGHT JOIN.

```
SELECT Nombre_Categoría, NombreProducto  
FROM Categorías INNER JOIN Productos  
ON Categorías.IDCategoría = Productos.IDCategoría;
```

CONSULTAS DE UNIÓN EXTERNAS

Se utiliza la operación UNION para crear una consulta de unión, combinando los resultados de dos o más consultas o tablas independientes. Su sintaxis es:

```
[TABLE] consulta1 UNION [ALL] [TABLE]  
consulta2 [UNION [ALL] [TABLE] consultan [ ... ]]
```

En donde:

consulta1, consulta2, consultan

Son instrucciones SELECT, el nombre de una consulta almacenada o el nombre de una tabla almacenada precedido por la palabra clave TABLE.

```
SELECT Nombre, Ciudad FROM Proveedores WHERE País = 'Brasil'  
UNION  
SELECT Nombre, Ciudad FROM Clientes WHERE País = "Brasil"
```



SENATI