

Esta clase va a ser
● grabada

Módulo 2

Agregación de datos para analizar

Temario



M1

Organizar, formatear y ajustar los datos

- Repaso proceso analítico de datos
- Fase de análisis
- Ordenar y filtrar

M2

Agregación de datos para analizar

- Concepto de agregación
- BUSCARV en Excel
- JOIN en SQL

M3

Realizar cálculos de datos

- Funciones condicionales
- Funciones escalares
- Tablas temporales

Agregar datos para análisis

Agregar datos para análisis

Proceso de recolectar datos de múltiples fuentes para combinarlos en una sola colección resumida

- Agregar contexto a los datos
- Hacer comparaciones
- Creación de campos y medidas calculadas

DATA



SORTED



ARRANGED



PRESENTED
VISUALLY



EXPLAINED
WITH A STORY



BUSCARV

BUSCARV o VLOOKUP

Relaciona datos de hojas distintas que tengan un dato en común.

Tener en cuenta previamente la **limpieza de los datos**

- Verificar que las columnas tengan el mismo **tipo de dato**
- Verificar que no contengan **espacios de más**
- Verificar que la tabla a relacionar no contenga **duplicados**

BUSCARV o VLOOKUP

Sintaxis

BUSCARV (valor_buscado; matriz_buscar_en; indicador_columnas; [ordenado])

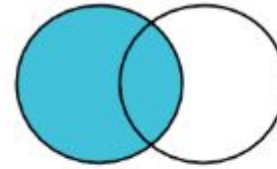
- **valor_buscado:** Puede ser un valor o una referencia a una celda
- **matriz_buscar_en:** Rango de celdas donde se busca el valor_buscado
- **indicador_columnas:** El número de columna que contiene el valor devuelto
- **ordenado:** Especifica si se desea buscar una coincidencia exacta o aproximada

JOIN

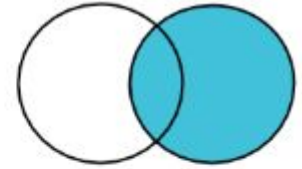
JOIN

Cláusula de SQL que se usa para combinar filas de dos o más tablas que cumplan con una condición.

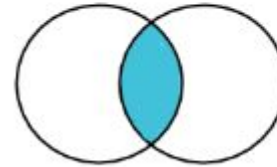
Es similar al **BUSCARV** de Excel



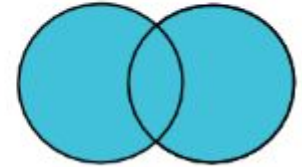
Left Join



Right Join



Inner Join



**Full Outer
Join**

INNER JOIN

Retorna todas las filas de las dos tablas siempre y cuando haya coincidencia por el campo declarado en el **ON**.

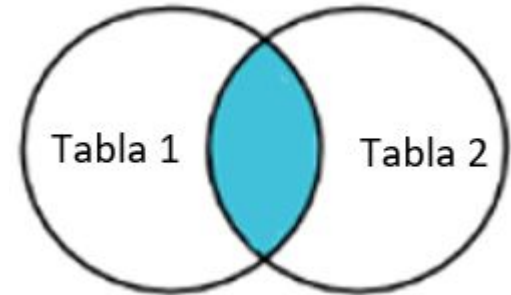
Sintaxis del Inner Join:

SELECT nombres_columnas

FROM tabla1

INNER JOIN tabla2

ON tabla1.columna_relacion=tabla2.columna_relacion;



INNER JOIN

TABLA 1		
nombre	id_departamento	puesto
Dave Smith	1	Gerente de Marketing de Producto
Julie Jones	2	Ingeniera de Software
Scott Tanner	1	Director de Generación de Demanda
Ted Connors	2	Ingeniera de Software
Margaret Lane	1	Vicepresidente de Marketing
Mary Martin	5	Recepcionista

TABLA 2	
nombre_dep	id_departamento
Marketing	1
Ingeniería	2
Contabilidad	3
Ventas	4

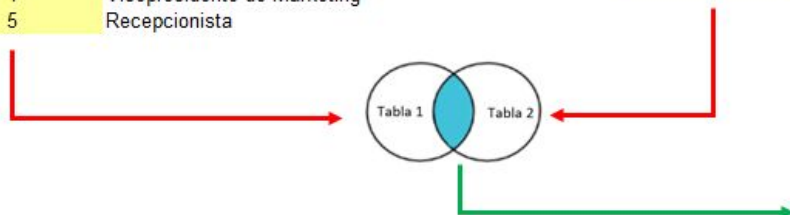


TABLA RESULTADO			
id_departamento	nombre	puesto	nombre_dep
1	Dave Smith	Gerente de Marketing de Producto	Marketing
1	Scott Tanner	Director de Generación de Demanda	Marketing
1	Margaret Lane	Vicepresidente de Marketing	Marketing
2	Julie Jones	Ingeniera de Software	Ingeniería
2	Ted Connors	Ingeniera de Software	Ingeniería

LEFT JOIN

Retorna **todas** las filas de la tabla **izquierda** (tabla1), con las filas **coincidentes** en la tabla **derecha** (tabla2).

El resultado es **NULL** en el lado derecho cuando no hay coincidencia.

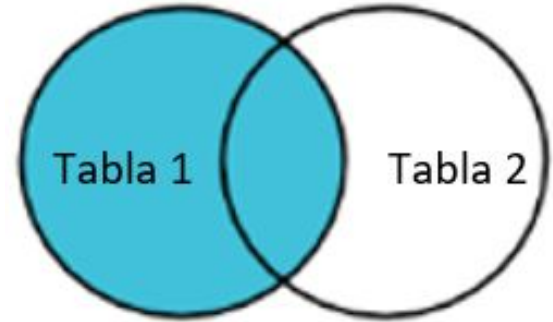
Sintaxis del Left Join:

SELECT nombres_columnas

FROM tabla1

LEFT JOIN tabla2

ON tabla1.columna_relacion=tabla2.columna_relacion;



LEFT JOIN

TABLA 1		
nombre	id_departamento	puesto
Dave Smith	1	Gerente de Marketing de Producto
Julie Jones	2	Ingeniera de Software
Scott Tanner	1	Director de Generación de Demanda
Ted Connors	2	Ingeniera de Software
Margaret Lane	1	Vicepresidente de Marketing
Mary Martin	5	Recepcionista

TABLA 2	
nombre_dep	id_departamento
Marketing	1
Ingeniería	2
Contabilidad	3
Ventas	4

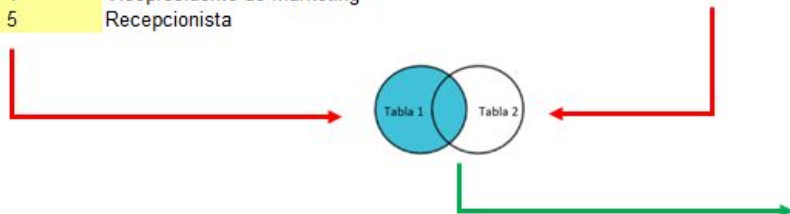


TABLA RESULTADO			
id_departamento	nombre	puesto	nombre_dep
1	Dave Smith	Gerente de Marketing de Producto	Marketing
2	Julie Jones	Ingeniera de Software	Ingeniería
1	Scott Tanner	Director de Generación de Demanda	Marketing
2	Ted Connors	Ingeniera de Software	Ingeniería
1	Margaret Lane	Vicepresidente de Marketing	Marketing
5	Mary Martin	Recepcionista	null

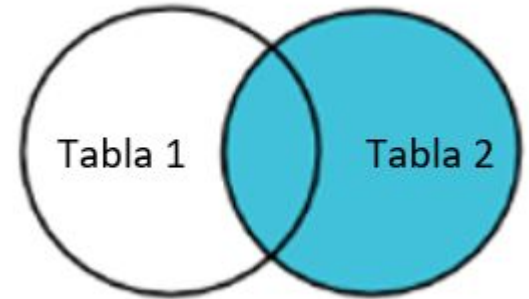
RIGHT JOIN

Devuelve **todas** las filas de la tabla **derecha** (tabla2), con las filas **coincidentes** en la tabla de la **izquierda** (tabla1).

El resultado es **NULL** del lado izquierdo cuando no hay coincidencia.

Sintaxis del Right Join:

```
SELECT          nombres_columnas
FROM            tabla1
RIGHT JOIN      tabla2
ON tabla1.columna_relacion=tabla2.columna_relacion;
```



RIGHT JOIN

TABLA 1		
nombre	id departamento	puesto
Dave Smith	1	Gerente de Marketing de Producto
Julie Jones	2	Ingeniera de Software
Scott Tanner	1	Director de Generación de Demanda
Ted Connors	2	Ingeniera de Software
Margaret Lane	1	Vicepresidente de Marketing
Mary Martin	5	Recepcionista

TABLA 2	
nombre dep	id departamento
Marketing	1
Ingeniería	2
Contabilidad	3
Ventas	4

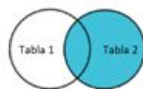


TABLA RESULTADO			
id departamento	nombre	puesto	nombre dep
1	Dave Smith	Gerente de Marketing de Producto	Marketing
2	Julie Jones	Ingeniera de Software	Ingeniería
1	Scott Tanner	Director de Generación de Demanda	Marketing
2	Ted Connors	Ingeniera de Software	Ingeniería
1	Margaret Lane	Vicepresidente de Marketing	Marketing
null	null	null	Contabilidad
null	null	null	Ventas

OUTER JOIN

Retorna todas las filas de la tabla de la **izquierda** (tabla1) y de la tabla de la **derecha** (tabla2).

El **OUTER JOIN** combina el resultado de **LEFT** y **RIGHT JOIN**. En este caso podemos tener valores **NULL** de ambos lados.

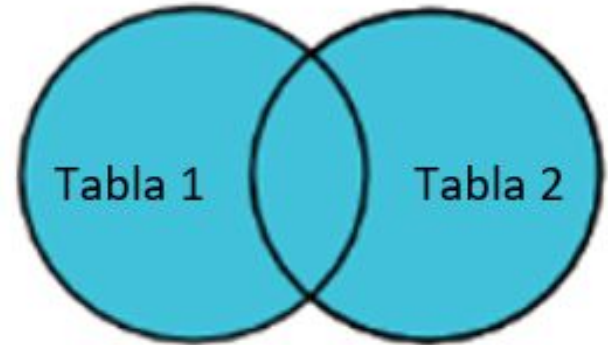
Sintaxis del FULL OUTER JOIN:

SELECT nombres_columnas

FROM tabla1

OUTER JOIN tabla2

ON tabla1.columna_relacion=tabla2.columna_relacion;



OUTER JOIN

TABLA 1		
nombre	id_departamento	puesto
Dave Smith	1	Gerente de Marketing de Producto
Julie Jones	2	Ingeniera de Software
Scott Tanner	1	Director de Generación de Demanda
Ted Connors	2	Ingeniera de Software
Margaret Lane	1	Vicepresidente de Marketing
Mary Martin	5	Recepcionista

TABLA 2	
nombre_dep	id_departamento
Marketing	1
Ingeniería	2
Contabilidad	3
Ventas	4

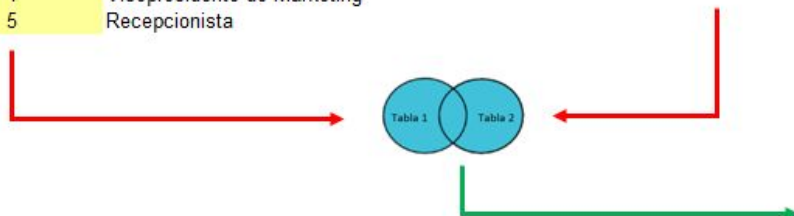


TABLA RESULTADO			
id_departamento	nombre	puesto	nombre_dep
1	Dave Smith	Gerente de Marketing de Producto	Marketing
2	Julie Jones	Ingeniera de Software	Ingeniería
1	Scott Tanner	Director de Generación de Demanda	Marketing
2	Ted Connors	Ingeniera de Software	Ingeniería
1	Margaret Lane	Vicepresidente de Marketing	Marketing
5	Mary Martin	Recepcionista	null
null	null	null	Contabilidad
null	null	null	Ventas

Alias

AS

Permite renombrar el resultado (de forma temporal) de cualquier objeto dentro de la base de datos: campos, tablas, vistas, entre otros.

Sintaxis:

```
SELECT nombres_columna AS alias_columna  
FROM tabla1
```

```
SELECT nombres_columna  
FROM tabla1 AS alias_tabla1
```

Funciones de agregación

COUNT y COUNT DISTINCT

Llamamos funciones de agregación a aquellas que **resumen** los resultados de una consulta (agregan datos).

- **COUNT:** Cuenta el número total de filas en un conjunto de datos.
- **COUNT DISTINCT:** Cuenta el número de valores únicos de una columna.

Sintaxis:

```
SELECT COUNT(*)
```

```
FROM tabla1
```

```
SELECT COUNT (DISTINCT nombre_columna)
```

```
FROM tabla1
```

GROUP BY

Las funciones de agregación devuelven un solo valor para toda la consulta.

Utilizando la cláusula **GROUP BY** podemos **aperturar** este resultado en varias filas, aportando contexto al análisis.

Sintaxis:

```
SELECT nombre_columna,  
        COUNT(*)  
FROM tabla1  
GROUP BY 1
```

Subqueries

Subqueries o subconsultas

Consiste en una sentencia **SELECT** dentro de otra sentencia **SELECT**.

- Permiten desarrollar consultas compuestas
- Pueden retornar un valor o una lista de valores.
- Pueden utilizar cualquiera de la cláusulas que vimos hasta ahora (DISTINCT, WHERE, ORDER BY, GROUP BY, etc.)

Sintaxis:

SELECT columna1, columna2

FROM tabla1

WHERE columna3 **IN** (**SELECT** columna4 **FROM** tabla2 **WHERE** condicion);

Demo guiada

¿Preguntas?

Resumen de la clase de hoy

- Aprendimos cómo relacionar tablas para sumar contexto al análisis.
- Revisamos la aplicación de las funciones de agregación.
- Revisamos la aplicación de subconsultas para análisis más complejos



**Recuerden realizar la
práctica de la clase**

¡Gracias!