



Universidad
Tecmilenio®





Consulta en Microsoft SQL Server®

Clasificación y filtrado de
datos

Semana 2



En este tema, analizarás los conceptos de información, datos y metadatos; comprenderás la importancia de clasificarlos para poder diseñar adecuadamente las tablas donde serán almacenados y poder recuperarlos de forma rápida y confiable, a través de filtros que aplicarás dentro de la cláusula **SELECT**.



Clasificación de datos



Los datos son cualquier palabra que signifique algo: nombres de personas, animales, cosas, lugares o países, acciones, eventos, alguna imagen, un número, etc.; estamos acostumbrados a referirnos a ellos como “información”, cuando en realidad son datos. Es importante entender que los datos “son la materia prima” para articular la información.

Metadatos



- Documento en el que incluye información como el autor de un libro, el tamaño del archivo, la fecha de creación del documento y alguna descripción del contenido del documento.

Otros comparadores

BETWEEN
IN
LIKE
IS NULL

```
SELECT * FROM T_Alumnos  
WHERE promedio  
NOT BETWEEN 80 AND 100
```

```
SELECT * FROM T_Alumnos  
WHERE promedio  
IN (70,80,90,100)
```

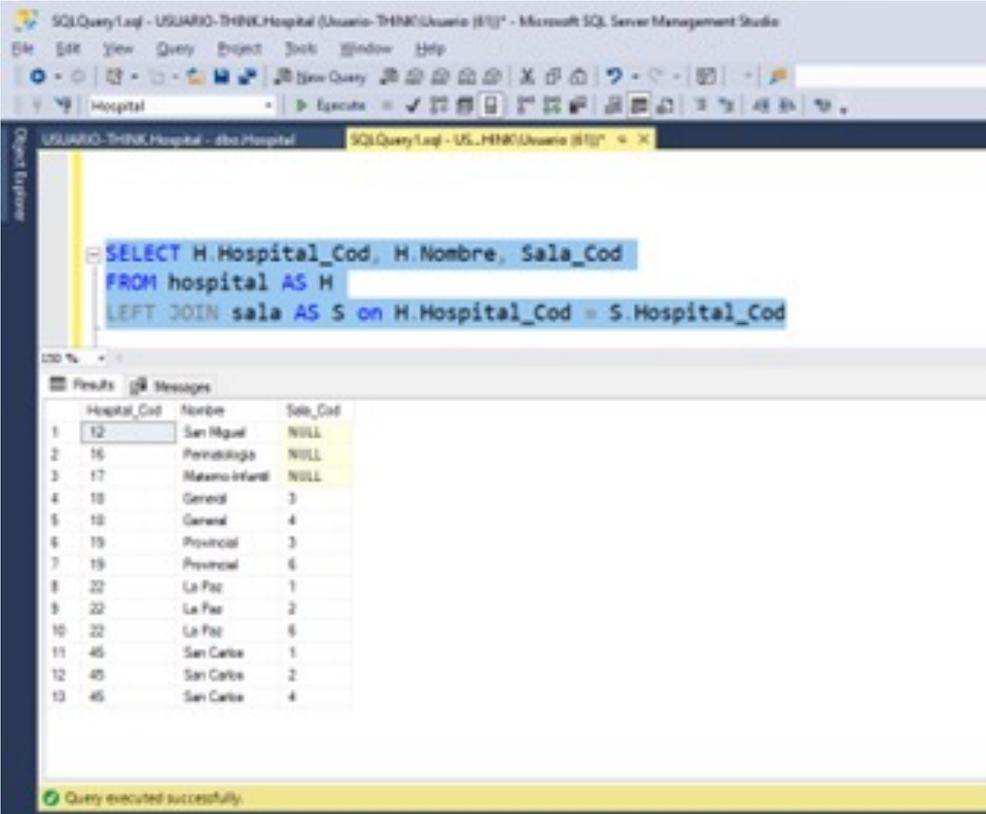
Valores desconocidos (NULL)

- EL valor NULL no es el equivalente a que valga CERO el registro, significa que no se encontró ningún valor para ese campo y por lo tanto está vacío o su valor es desconocido (MySQL, 2022).

```
151 $result1 = mysql_query($sql17);  
152 $result2 = mysql_query($sql12);  
153 $result3 = mysql_query($sql13);  
154 $result4 = mysql_query($sql14);  
155 $result5 = mysql_query($sql15);  
156 $result6 = mysql_query($sql16);  
157  
158 $result11 = mysql_query($sql111);  
159 $result22 = mysql_query($sql122);  
160 $result33 = mysql_query($sql133);  
161 $result44 = mysql_query($sql144);  
162 $result55 = mysql_query($sql155);  
163 $result66 = mysql_query($sql166);  
164  
165 mysql_close();  
166  
167  
168 echo '<p style="font-size:12px; font-family:Arial, Helv  
169  
170 if (
```

Para analizar

```
SELECT H.Hospital_Cod, H.Nombre, Sala_Cod  
FROM hospital AS H  
LEFT JOIN sala AS S on H.Hospital_Cod = S.Hospital_Cod
```



The screenshot shows the Microsoft SQL Server Enterprise interface. The query editor contains the following SQL code:

```
SELECT H.Hospital_Cod, H.Nombre, Sala_Cod  
FROM hospital AS H  
LEFT JOIN sala AS S on H.Hospital_Cod = S.Hospital_Cod
```

The Results pane displays the following data:

	Hospital_Cod	Nombre	Sala_Cod
1	12	San Miguel	NULL
2	16	Perinologica	NULL
3	17	Maximo Infant	NULL
4	18	General	3
5	18	General	4
6	19	Provincial	3
7	19	Provincial	6
8	22	La Paz	1
9	22	La Paz	2
10	22	La Paz	6
11	45	San Carlos	1
12	45	San Carlos	2
13	45	San Carlos	4

Query executed successfully.

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01

¿Conocías los metadatos?, Por lo que has visto en el tema, ¿puedes dar un ejemplo de un metadato?

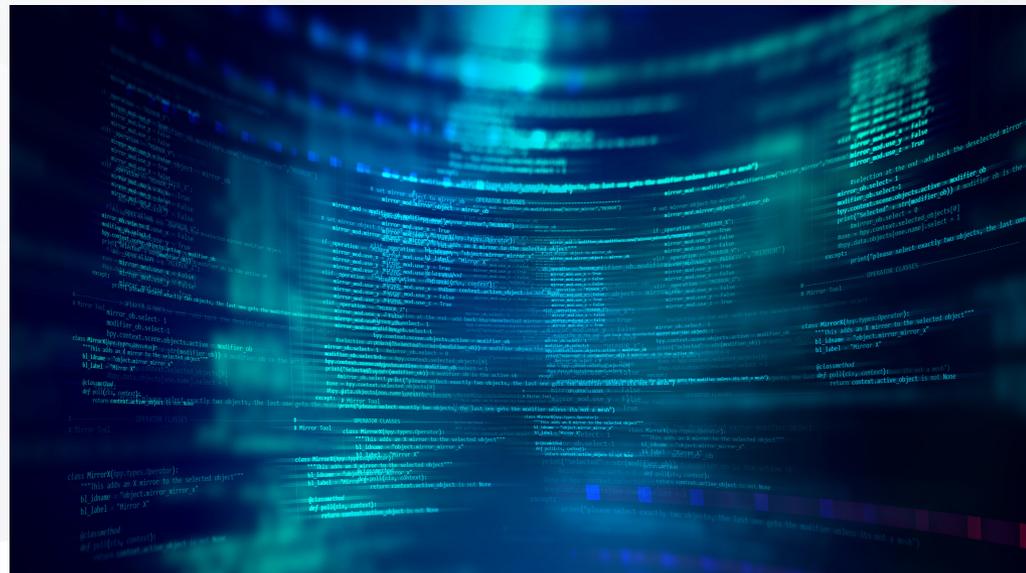
02

¿Qué operador de filtrado para WHERE reconoces que es el más usado y por qué?



Todo **programador** de bases de datos debe conocer el significado de Datos, Información y Metadatos; es común referirse a datos e información como lo mismo, sin embargo, es fundamental saber la diferencia a la hora de diseñar una base de datos.

Los metadatos son un concepto que hoy en día toma mucha relevancia; el manejo de metadatos se vuelve fundamental para el correcto funcionamiento de estas aplicaciones.





Bibliografía

- MySQL. (2022). *MySQL 8.0 Reference Manual*. Recuperado de <https://dev.mysql.com/doc/refman/8.0/en/>



Consulta en Microsoft SQL Server®

Tipos de datos

Semana 2



Introducción a los tipos de datos

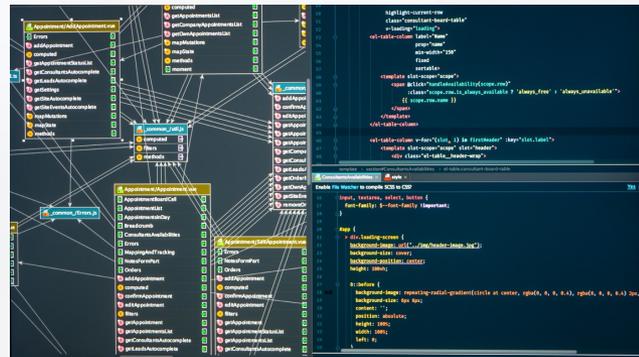
Los tipos de datos en programación son la característica o atributos que los “contenedores” de datos deberán considerar al almacenar en ellos valores; por ejemplo, en una tabla con los campos: nombre, edad y fecha de nacimiento, los datos que se resguardan en diferente forma: nombre utiliza texto, edad son números, fecha de nacimiento almacena formatos de fecha (Microsoft, 2022).

Asignar un tipo de dato a un campo de una tabla:

nombre_del_campo **AS** tipo_de_dato

Asignar un tipo de dato a una variable:

Declare @vNombre **AS** **varchar**(25),
 @vEdad **AS** **int**,
 @vFecha_de_nacimiento **AS** **DATE**
 @vSueldo **AS** **numeric**(8,2)



Datos de caracteres

A los tipos de datos de caracteres, se les conoce también como tipos de datos de cadenas, y están compuestos por letras, números y símbolos.

Char(n)	Tipo de datos de cadena de longitud fija. Almacena máximo 8000 caracteres.
Varchar(n)	Tipo de datos de cadena de longitud variable. Almacena máximo 8000 caracteres.

Los datos de tipo cadena en SQL Server se deben escribir entre comillas sencillas o dobles. Por ejemplo:

```
DECLARE @vNombre AS nvarchar(15)  
SET @vNombre = 'HOLA MUNDO'
```

Datos de fecha y hora

Para almacenar valores de fecha y hora existen en SQL tipos de datos que permiten hacerlo; depende del formato que se desee almacenar del valor. En SQL existen los siguientes tipos:



- DATETIME
- DATETIME2
- DATE
- SMALLDATETIME
- TIME

Ejemplo de datos de fecha y hora

```
DECLARE @vFechaAcual AS date -- se declara la variable vFechaActual  
de tipo date  
SET @vFechaAcual = '2022/05/23' -- se asigna el valor a la variable  
vFechaActual
```

```
DECLARE @vFechaHora AS datetime - - se declara la variable  
vFechaHora de tipo datetime  
SET @vFechaHora = '2022/05/23 01:59:16' - - se asigna el valor a la  
variable vFechaHora
```

```
DECLARE @vHora AS time -- se declara la variable vHora de tipo  
time  
SET @vHora = '01:59:16' -- se asigna el valor a la variable vHora
```



Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01

¿Cuál sería la gravedad y afectaciones que tendría un sistema que toda la definición de tipo de datos fuera de tipo varchar(n)?

02

¿Cómo explicar la diferencia entre definir DATETIME y DATETIME2?





En una base de datos de poco tamaño no es tan relevante el correcto manejo de los tipos de datos, es decir, puedes usar un *int* o un *bigint* para almacenar un entero y lo permitirá, o un *char* y un *varchar*, sin embargo, en las grandes bases de datos esto atrofia el funcionamiento de los sistemas, el servidor se ralentiza, lo que provoca que los usuarios finales tengan una mala experiencia utilizando el sistema informático.

Bibliografía



- Microsoft. (2022). *Tipos de datos (Transact-SQL)*. Recuperado de: <https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15>



Consulta en Microsoft SQL Server®

Introducción al lenguaje T-SQL

Semana 2



Hoy en día las organizaciones manejan grandes cantidades de datos, los cuales son útiles siempre y cuando se le dé un uso en tiempo y forma. Los datos por sí mismo son irrelevantes, si no se les da un uso, sentido o propósito. A lo largo de este tema se tocan los conceptos más importantes para tener un **óptimo manejo y extracción de datos**.



Introducción al lenguaje T-SQL



La diferencia entre SQL y T-SQL es que T-SQL es una extensión de SQL e incluye más elementos y comandos de los incluidos en SQL.

T-SQL acrónimo en inglés de "Transact Structured Query Language", lenguaje de consulta utilizado en Microsoft SQL Server.

El lenguaje de SQL es más usado para fines de manipulación de datos.

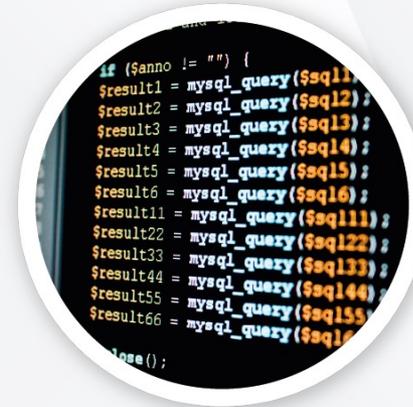
El lenguaje de T-SQL se utiliza para creaciones de funciones avanzadas y aplicaciones.



- SQL se usa para crear consultas, que permiten extraer datos e información de manera estructurada.
- T-SQL, puede crear aplicaciones relacionadas con bases de datos, como lógica para el funcionamiento de un proceso de negocio.

Funciones agregadas

En T-SQL se puede agregar la función de DISTINCT a otras funciones.



Palabra clave de *Output*

Cuando se requiere actualizar algún valor en específico para toda una tabla, se aplica la cláusula *Output*, para que el registro se realice adecuadamente.



Alterar la tabla con la función JOIN

Es importante mencionar este ejemplo dada la importancia que ella tiene en el lenguaje SQL.



Estructura de las instrucciones de SQL

T-SQL cuenta con sus propios comandos preestablecidos que están pensados para ejecutarse de manera estructurada en SQL (Microsoft, s.f.).

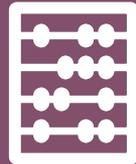
Clausula SQL	Función
SELECT	Muestra la lista de los campos o columnas de una tabla a seleccionar.
FROM	Se especifican las tablas de los cuales el SELECT tomará campos/columnas.
WHERE	Especifica condiciones de campo que cada registro debe cumplir para poder ser incluido en los resultados.
ORDER BY	Especifica la forma de ordenar los resultados. (Ascendente y Descendente).
GROUP BY	Agrupar los resultados de la cláusula SELECT en base a un campo como referencia.
HAVING	Extiende el filtrado de la cláusula WHERE para funciones agregadas como la de Group By.

Tipos de datos

Los lenguajes de programación tales como Python y Java, existen tipos de datos, que son necesarios tener presentes en las bases de datos. Naturalmente los datos que se manejan dentro de SQL varían.



DATOS DE TEXTO



DATOS NUMÉRICOS



DATOS DE TIEMPO

Ordenado y filtrado de datos

En una base de datos es necesario un buen uso a través de un correcto ordenamiento y filtrado de los datos. Normalmente se hace con la finalidad de incluir filas y campos que cumplan con criterios específicos.



```
SELECT <Campo/s o *>  
FROM <Fuente (Tabla)>  
ORDER BY <ordenar la lista de resultados> [ASC|DESC].
```



```
SELECT TOP (N) <Lista de Columnas>  
FROM <Tabla fuente>  
WHERE <Condición>
```



```
SELECT DISTINCT <Lista de Columnas>  
FROM <Tabla fuente>  
ORDER BY <Lista de Columnas>
```

Filtrado de datos

Existen varias maneras de condicionar el filtrado de datos. El comando **WHERE** es por naturaleza el comando utilizado para filtrar datos. Para filtrar los datos de la manera que el usuario desee, es necesario aprender una serie de condiciones que se pueden aplicar (Microsoft, 2022).

- = (es igual a)
- <> (no es igual a)
- > (mayor que)
- >= (mayor o igual que)
- < (menor que)
- <= (menor o igual que)



```
SELECT IDservicio
FROM ServicioTabla
WHERE IDservicio = 2;
```

```
SELECT Precio
FROM ProductoTabla
WHERE Precio < 10;
```

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01

¿En qué escenario se aplicaría un TOP en T-SQL?

02

Si se desea obtener a los 10 mejores alumnos de una institución, ¿Qué comandos se deberían aplicar?





A lo largo de este tema se revisaron comandos claves que ofrece el código de T-SQL para el ambiente de SQL. Estos comandos son básicos y fundamentales para lograr una buena estructura; así como la extracción y el filtrado de datos. Por otro lado, se revisaron los tipos de datos más comunes dentro del lenguaje.

Cierre





Bibliografía

- Microsoft. (s.f.). *Access SQL: conceptos básicos, vocabulario y sintaxis*. Recuperado de <https://support.microsoft.com/es-es/office/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671>
- Microsoft. (2022). *Ordenación y filtrado de resultados en T-SQL*. Recuperado de [https://docs.microsoft.com/es-es/learn/modules/sort-filter-queries/enación y filtrado de resultados en T-SQL - Learn | Microsoft Docs](https://docs.microsoft.com/es-es/learn/modules/sort-filter-queries/enación-y-filtrado-de-resultados-en-T-SQL-Learn-Microsoft-Docs)



Consulta en Microsoft SQL Server®

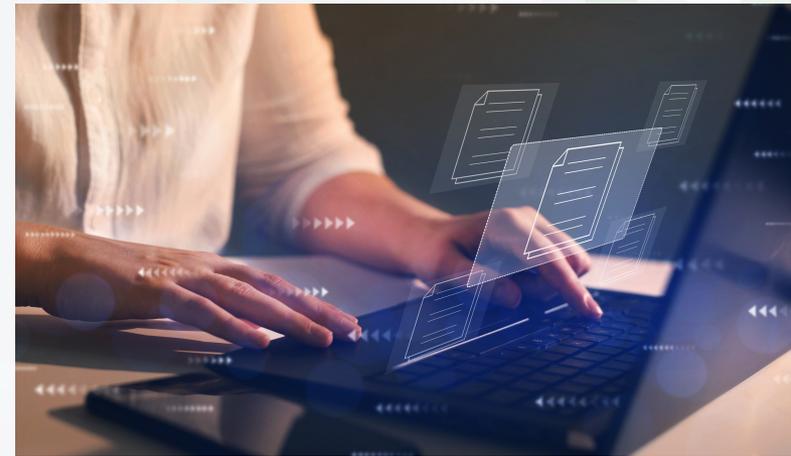
Combinación de tablas por
medio de JOINS

Semana 2



En el lenguaje de T-SQL existe una cláusula que es utilizada para combinar diferentes datos de diferentes maneras. Las organizaciones hoy en día necesitan de expertos que no necesariamente creen nuevas tecnologías, sino que sepan usar las ya existentes y generar impactos positivos y tangibles a través de ellas.

Una cosa es saber manejar, extraer, ordenar y filtrar datos de manera básica, haciendo uso de cláusulas básicas como SELECT y WHERE. Pero, existen más cláusulas dentro del lenguaje de T-SQL tal como la cláusula JOIN. Esta cláusula es muy conocida y si su aplicación se hace correctamente, se puede tener una mejor manipulación de datos.



Combinación de tablas por medio de JOINS

La cláusula **JOIN** existe para poder lograr combinaciones entre dos diferentes tablas **siempre y cuando tengan una columna en común**.

- Logra traer los datos de dos columnas y tablas diferentes.
- Existen cuatro tipos que sirven de base para las posibles combinaciones de JOIN's (Microsoft, 2021):

- **INNER JOIN**
- **LEFT JOIN**
- **RIGHT JOIN**
- **FULL OUTER JOIN**

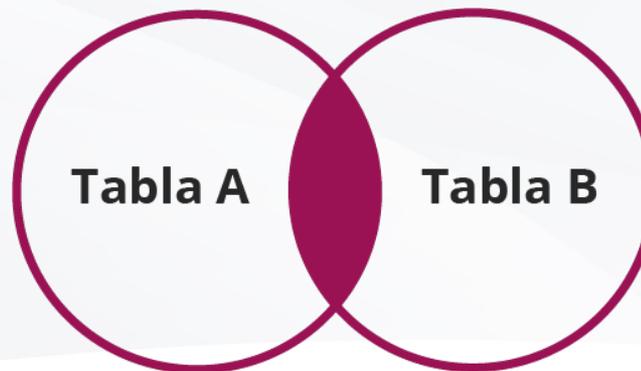


Combinaciones internas

Inner Join

Esta combinación es uno de los tres tipos de combinaciones que existen. Es considerada como la más básica y fundamental, ya que mantiene la premisa de hacer una unión entre dos tablas. Existen dos tipos de sintaxis para el único tipo de combinación interna que existe.

```
SELECT TablaA.Ciudad, TablaA.Nombre, TablaB.Apellido  
FROM TablaA  
INNER JOIN TablaB ON TablaA.Ciudad=TablaB.Ciudad;
```

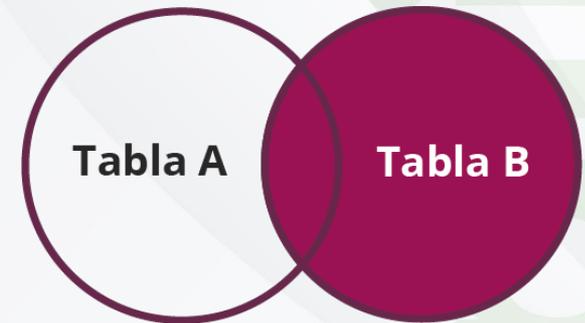


Combinaciones externas

Right Join

Busca traerse todos los registros de la tabla B, sin importar, si algunos de estos no empatan con los de la tabla A. Asimismo, se trae los registros similares entre la tabla A y B.

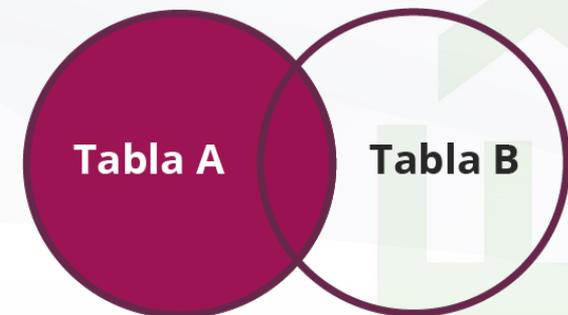
```
SELECT Lista de Columnas  
FROM TablaA  
RIGHT JOIN TablaB ON Condicion del Join.
```



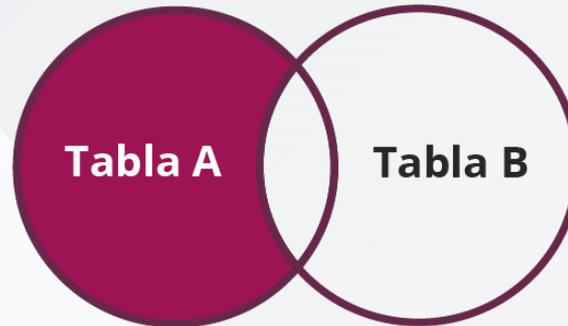
Left Join

Busca traerse todos los registros de la tabla A (izquierda), sin importar, si algunos no empatan con los de la tabla B (derecha). Además, se trae los registros similares entre la tabla A y B.

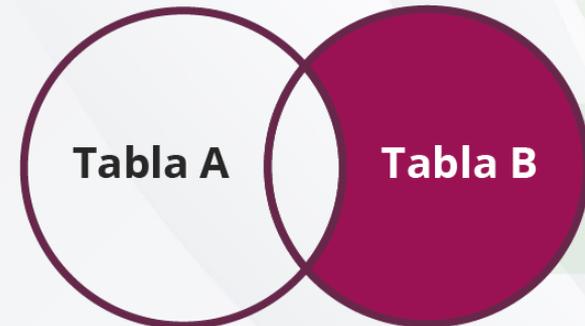
```
SELECT Lista de Columnas  
FROM TablaA  
LEFT JOIN TablaB ON Condicion del Join.
```



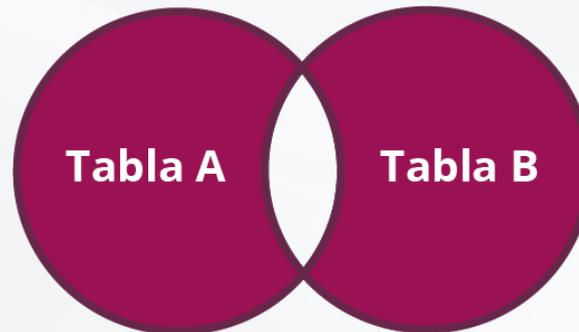
Se muestran más JOINS



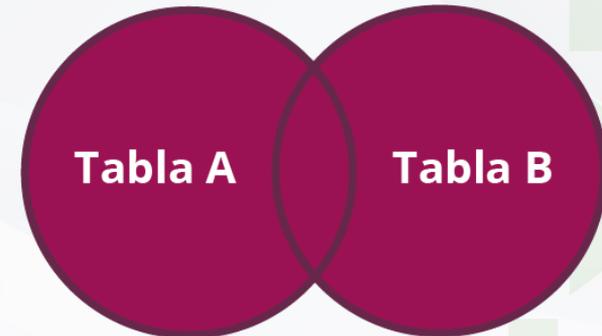
Right Outer Join



Left Outer Join



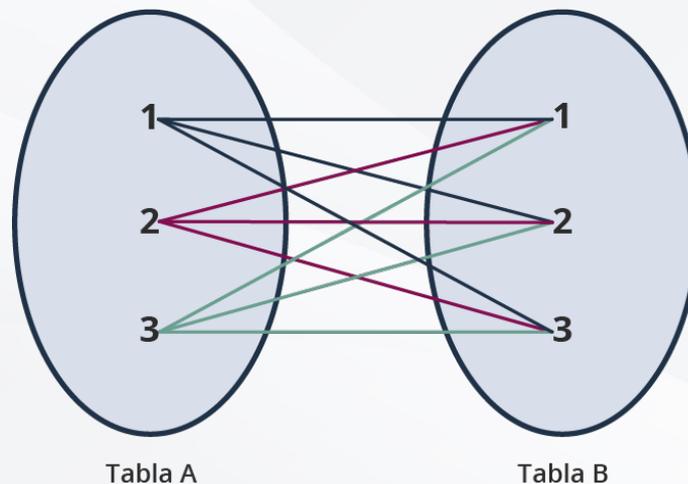
Outer excluding Join



Full Outer Join

Combinaciones cruzadas

La aplicación de la sintaxis de este JOIN es sumamente sencilla, ya que entrelaza columnas y filas. La correcta aplicación de esta consulta también recae en las columnas seleccionadas para entrelazar. Al final, recae en el criterio del analista de datos, escoger aquellas columnas entrelazadas con otras y con todos sus campos puedan reflejar algún tipo de significancia para el ejercicio en curso.



```
SELECT Lista de Columnas  
FROM TablaA  
Cross Join TablaB
```



Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01

¿Qué importancia tiene que la información se encuentre relacionada en una base de datos?

02

Los JOINS son sentencias para unir dos o más tablas para recolectar información, ¿de qué manera se puede optimizar los recursos en este tipo de consultas?



En el mundo práctico, la correcta aplicación de esta cláusula en relación con las tablas y campos en estudio permite obtener aquellos datos que son relevantes para cualquiera que sea el ejercicio en curso.

Distinguir las diferentes aplicaciones de la cláusula JOIN permiten al usuario extraer datos que sean objeto de estudio. Lo más importante, es entender cómo la aplicación correcta de estas funciones nos puede llevar a un análisis de datos adecuados.





Bibliografía

- Microsoft. (2021). *Combinaciones (SQL Server)*. Recuperado de <https://docs.microsoft.com/es-es/sql/relational-databases/performance/joins?view=sql-server-ver15>
- PCSoft. (2022). *SQL: Managing joins (INNER JOIN, OUTER JOIN)*. Recuperado de <https://doc.windev.com/en-US/?2034007>