

Esta clase va a ser

- grabada

a



Clase en vivo

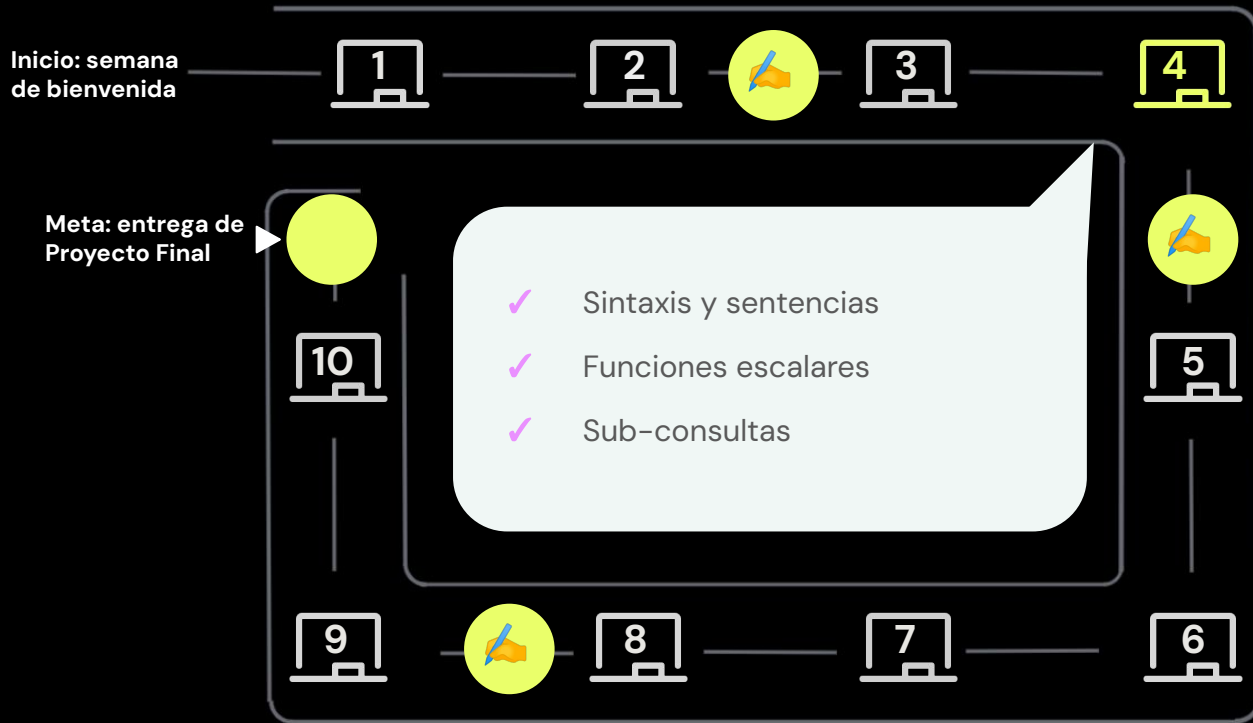
¿Comenzamos?

Lenguaje SQL

¿Preguntas?

Te invitamos a dejar tu
pregunta en el [chat de](#)
[Zoom](#)

Roadmap



Objetivos

- **Realizar** una puesta en común del 4º microdesafío: Consulta de datos, funciones y agrupaciones.
- **Ejercitar** select y from.
- **Diseñar** un diagrama de entidad-relación y las tablas correspondientes.
- **Ejercitar** distinct, where, as, order by, limit, group by y having.



REPASO

Lenguaje SQL, funciones escalares y subconsultas

Esta semana, a través del material on demand...

- ✓ Conocimos las características y sintaxis del lenguaje SQL
- ✓ Conocimos las sentencias más comunes de SQL
- ✓ Manipulamos datos con sentencias más básicas
- ✓ Generamos transformaciones de datos con funciones
- ✓ Identificamos e implementamos subconsultas



Puesta en común microdesafío

¡Vamos a recuperar lo trabajado durante la semana! 🚀

Duración: **15 minutos.**



PUESTA EN COMÚN – MICRODESAFÍO

Consulta de datos, funciones y agrupaciones

[Acceso a la consigna completa](#) 🚀



CODERHOUSE



PUESTA EN COMÚN – MICRODESAFÍO

Ejercicio 1

¿Qué aprendimos?

`/* Comando select distinct*/`

```
SELECT DISTINCT Ciudad,ID_Estado  
FROM [dbo].[Ciudades_USA];
```

Cuando utilizamos DISTINCT quitamos los duplicados según la combinación de columnas.





PUESTA EN COMÚN – MICRODESAFÍO

Ejercicio 2

¿Qué aprendimos?

```
SELECT *  
FROM [dbo].[Ciudades_USA]  
WHERE ID_ESTADO = 5;
```

Utilizando Where y operadores podemos filtrar la información que obtenemos de nuestras consultas.



CODERHOUSE



PUESTA EN COMÚN – MICRODESAFÍO

Ejercicio 3

¿Qué aprendimos?

```
SELECT *  
FROM [dbo].[Ciudades_USA]  
WHERE Ciudad LIKE 'M%'  
OR ID_ESTADO > 30;
```

Nuestro resultado se reduce a menos de la mitad que la consulta anterior, esto es por el uso de más de 2 condicionales.



CODERHOUSE



PUESTA EN COMÚN – MICRODESAFÍO

Ejercicio 4

¿Qué aprendimos?

- En **order by únicamente** puedo usar el nombre que asignamos como nickname mediante el AS.
- En esta ocasión se ordena por la segunda columna, que es CANTIDAD.

```
SELECT ID_Estado, COUNT(*) AS CANTIDAD  
FROM [dbo].[Ciudades_USA]  
GROUP BY ID_Estado  
ORDER BY Cantidad DESC;
```



CODERHOUSE

SQL

Operadores



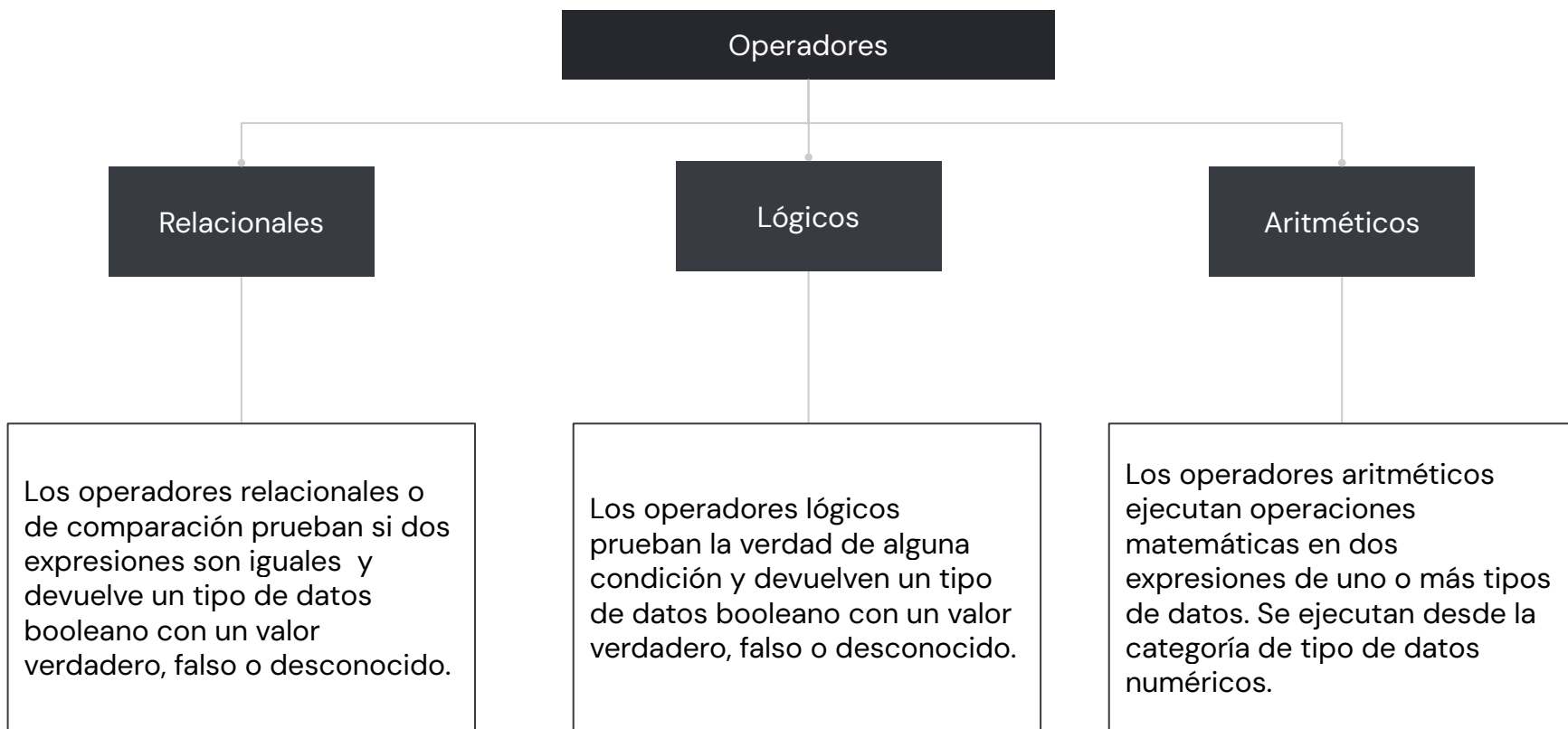
VIDEO N°4.3 – Hablando SQL

Operadores

En el microdesafío trabajamos sobre las sentencias más comunes de SQL.

Ahora retomemos los operadores que es un símbolo que especifica una acción que se realiza en una o más expresiones de las consultas que utilicemos en SQL.







Operador	Condición	Ejemplo
=	Verdadero si es igual	where campo = "valor"
!=	Verdadero si es diferente	where campo != "valor"
>	Verdadero si es mayor que	where campo > "valor"
<	Verdadero si es menor que	where campo < "valor"
>=	Verdadero si es mayor e igual que	where campo >="valor"
<=	Verdadero si es menor e igual que	where campo <="valor"

Operador	Condición	Ejemplo
AND	Verdadero si se cumplen dos o más condiciones	WHERE condicion1 AND condicion2 AND condicion3...
OR	Verdadero si se cumple alguna de las condiciones	WHERE condicion1 OR condicion2 OR condicion3...
NOT	Verdadero si no se cumple la condición	WHERE NOT condición
LIKE	Verdadero si cumple un patrón	WHERE columna1 LIKE patrón
IN	Verdadero si encuentra alguno de los valores otorgados	WHERE columna1 IN (valor1, valor2, valor3...)
BETWEEN	Verdadero si el valor se encuentra en el rango	WHERE columna1 BETWEEN valor1 AND valor2

Operador	Operación	Ejemplo
+	Permite sumar los valores de dos columnas, una columna y un valor, o dos valores.	Select campo1 + campo2 as campo3 ...
-	Permite restar .	Select campo1 - campo2 as campo3 ...
*	Permite multiplicar .	Select campo1 * campo2 as campo3 ...
/	Permite dividir .	Select campo1 / campo2 as campo3 ...



Para pensar

Identifica cuál es el error en las siguientes consultas:

```
SELECT DISTINCT Nombre, Telefono
FROM Clientes
WHERE N_Cliente > '400';
```

```
SELECT Nombre, Telefono
FROM Clientes
WHERE Nombre = Mariela AND Sexo = F
```

```
SELECT Sexo, Nombre
COUNT (N_Cliente) AS Cant_Clientes
FROM CLIENTES
GROUP BY Sexo, Nombre
WHERE COUNT (N_Cliente) < 400 ;
```

```
SELECT Sexo, Nombre, Telefono
FROM CLIENTES
WHERE Nombre LIKE 'Fernando';
```

Contesta mediante el chat de Zoom.

Tabla para ejercicios

CREATE SIN
AUTOINCREMENTAL

```
CREATE TABLE
Clientes(N_Cliente  int NOT
NULL PRIMARY KEY,
Nombre varchar(20) NOT NULL,
Sexo varchar(1) NOT NULL,
Telefono varchar(10) NOT NULL,
ID_Edad int NOT NULL);
```

INSERT

```
INSERT INTO Clientes
VALUES (345,'Juan', 'M','34484056',8 ),
(390,'Mariela', 'F','45835000',9 ),
(398,'Cristian', 'M','45786349',1 ),
(562,'Fernando', 'M','42568360',3 ),
(610,'Fernando', 'M','42568360',3 ),
(817,'Silvana', 'F','32583159',5 );
```



Actividad de repaso

Observa las siguientes consultas y determina qué registros traerá cada una:

- ✓ `SELECT * FROM Clientes;`
- ✓ `SELECT N_Cliente, Nombre
FROM Clientes;`
- ✓ `SELECT Nombre, Sexo
FROM Clientes;`
- ✓ `SELECT N_Cliente, Telefono, ID_Edad
FROM Clientes`

Contesta mediante el chat de Zoom



Diseño de un diagrama de entidad-relación y definición de tablas para tu proyecto final

Duración: 30 minutos



ACTIVIDAD

1º PARTE

Consigna:

- ✓ Diseñar el modelo entidad-relación de la temática elegida para el proyecto final.

Aspectos a incluir:

- ✓ Definir al menos cinco tablas.
- ✓ Crear el diagrama de entidad-relación con todos sus componentes:
 - Entidades
 - Acciones de relacionamiento
 - Tipos de relación
 - Campos clave

Formato:

- ✓ Formato: PPT, GSlides o Mural.

Sugerencias

- ✓ En caso de ser un archivo en línea, activar permisos de acceso.

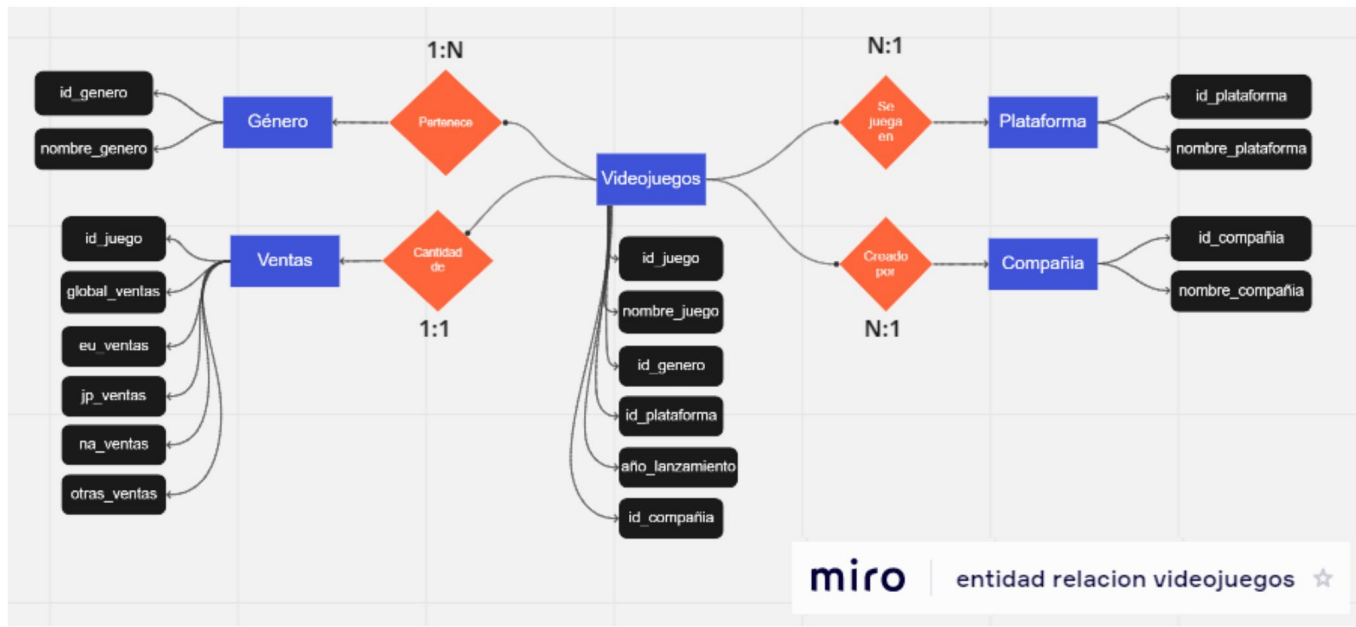
Duración: **15 minutos**



ACTIVIDAD

Ejemplo:

- VideosJuegos:





Puesta en común

Duración: 5 minutos



PUESTA EN COMÚN - ACTIVIDAD

Diseño de un diagrama de entidad-relación y definición de tablas para tu proyecto final

1º parte

- ✓ Verificar que todas las tablas tengan PK.
- ✓ Verificar que las PK sean únicas y NO nulas.
- ✓ Verificar que las tablas que figuran relacionadas tengan un campo en común (FK).



ACTIVIDAD

2° PARTE

Consigna:

- ✓ Basado en el modelo de entidad relación que se acaba de diseñar, crear cinco columnas por tabla.

Aspectos a incluir:

- ✓ Identificar las claves primarias, en todas las tablas.
- ✓ Identificar las claves candidatas, cuando corresponda.
- ✓ Identificar las claves foráneas.
- ✓ Identificar los tipos de datos de cada columna.

Formato:

- ✓ Formato: PPT, GSlides o Mural.

Sugerencias

- ✓ En caso de ser un archivo en línea, activar permisos de acceso.

Duración: **15 minutos**



ACTIVIDAD

Ejemplo:

DATASET 1: VIDEOJUEGOS

Tabla1: Videojuegos

Tipo de clave	Campo	Tipo de Campo
PK - index	Id_juego	int
-	Nombre_juego	nvarchar(100)
FK	Id_genero	Int
FK	Id_plataforma	Int
-	Ano_lanzamiento	datetime
FK	Id_compania	Int

Tabla2: Plataforma

Tipo de clave	Campo	Tipo de Campo
PK - index	Id_plataforma	int
-	Nombre_plataforma	nvarchar(100)

Tabla3: Compania

Tipo de clave	Campo	Tipo de Campo
PK - index	Id_compania	int
-	Nombre_compania	nvarchar(100)

Tabla4: Genero

Tipo de clave	Campo	Tipo de Campo
PK - index	Id_genero	int
-	Nombre_genero	nvarchar(100)



Puesta en común

Duración: 5 minutos



PUESTA EN COMÚN - ACTIVIDAD

Diseño de un diagrama de entidad-relación y definición de tablas para tu proyecto final

2ª parte

- ✓ Verificar tipos de datos de cada una de las columnas y que se encuentren alineados con su tipo de datos.
- ✓ Verificar que los tipos de datos de cada columna sean específicos en cuanto al límite de caracteres adquiridos.



Break

¡10 minutos y volvemos!

Sentencias SQL

DISTINCT – WHERE – AS



Ejemplo en vivo

Ejercitando con DISTINCT – WHERE – AS

Observemos las siguientes consultas y los registros que traerá cada una:



Ejemplo en vivo

1. `SELECT DISTINCT * FROM Clientes;`

1. `SELECT Nombre AS Cliente
FROM Clientes
WHERE ID_Edad = 3;`

3. `SELECT DISTINCT
Nombre AS Cliente,
Sexo,
Telefono
FROM Clientes
WHERE N_Cliente > 400;`

N. Cliente	Nombre	Sexo	Telefono	ID_EDAD
345	Juan	M	34484056	8
390	Mariela	F	45835000	9
398	Cristian	M	45786349	1
562	Fernando	M	42568360	3
610	Fernando	M	42568360	3
817	Silvana	F	32583159	5

ORDER BY – TOP
GROUP BY – HAVING



Ejemplo en vivo

**Ejercitando con ORDER BY – TOP – GROUP BY –
HAVING**

Observemos las siguientes consultas y los registros que traerá cada una:

Ejemplo en vivo

- `SELECT * FROM Clientes ORDER BY N_cliente desc;`
- `SELECT TOP(3) * FROM Clientes ORDER BY N_Cliente;`
- `SELECT COUNT (N_cliente)as Cantidad, nombre FROM Clientes GROUP BY nombre ORDER BY COUNT (N_CLIENTE) DESC;`
- `SELECT COUNT (N_cliente)as Cantidad, nombre FROM Clientes GROUP BY nombre HAVING COUNT (N_cliente) >= 2 ORDER BY COUNT (N_CLIENTE) DESC;`

N. Cliente	Nombre	Sexo	Telefono	ID_EDAD
345	Juan	M	34484056	8
390	Mariela	F	45835000	9
398	Cristian	M	45786349	1
562	Fernando	M	42568360	3
610	Fernando	M	42568360	3
817	Silvana	F	32583159	5

Funciones Escalares



VIDEO N°4.5 – Funciones escalares

Funciones escalares

Las funciones escalares son transformaciones que se aplican a los datos de nuestra base , pero únicamente de forma temporal en la vista que generamos a partir del SELECT.



Retomemos el ejemplo del video.

Una empresa de Japón que realiza envíos a países de latinoamérica necesita crear 2 tablas: una de Envíos y una de Estados que se relacionen entre sí.

CREATE

```
Create table
Estados(IDEstado int
NOT NULL IDENTITY
PRIMARY KEY,
Nombre varchar(50) NOT
NULL);
```

INSERT

```
Insert into Estados(Nombre)
values ('Ingresado en Planta'),('En
Camino'),('Espera en sucursal'),('Entregado');
```

YEAR, MONTH, DAY

```
SELECT YEAR(FechaEnvio) FROM
Envios;

SELECT MONTH(FechaEnvio)
FROM Envios;

SELECT DAY(FechaEnvio) FROM
Envios;

SELECT IDEnvio,
YEAR(FechaEnvio) as Anio,
MONTH(FechaEnvio) as Mes,
DAY(FechaEnvio) as Dia FROM
Envios;
```

DATENAME / DATEPART

```
SELECT IDEnvio,
DATENAME(MONTH, FechaEnvio
) as Mes FROM Envios;
```

DATEADD

```
SELECT IDEnvio,
DATEADD(MONTH,1, FechaEnvi
o) as Mes FROM Envios;
```

DATEDIFF

```
SELECT IDEnvio,
DATEDIFF(MONTH,
'05-17-2021', GETDATE())
as Diferencia FROM
Envios;
```

LEFT

```
SELECT *FROM Envios
WHERE LEFT (IDENVIO,2)='EX';

SELECT *FROM Envios
WHERE LEFT (IDENVIO,2)='EN';
```

RIGHT

```
SELECT *FROM Envios
WHERE RIGHT (IDENVIO,2)='AR';

SELECT *FROM Envios
WHERE RIGHT
(IDENVIO,2)='CO';
```

CONCAT

```
SELECT CONCAT(IDEnvio, IDEstado) AS Tracking
FROM Envios;

SELECT
CONCAT(LEFT (IDEnvio,2), (RIGHT (IDEnvio,2)))
AS TipoEnvio, IDEstado, Descripcion FROM
Envios;
```

REPLACE

```
SELECT REPLACE(IDEnvio, '-', ' ')FROM Envios;

SELECT
REPLACE(REPLACE(REPLACE(REPLACE(REPLACE (IDEn
vio, 'AR', ' Argentina'), 'CL', ' Chile'),
'UY', 'Uruguay'), 'CO', 'Colombia'), 'MX',
'Mexico') FROM Envios;
```

CODER HOUSE

CODERHOUSE



Práctica SQL

Consultas SQL con funciones especiales y subconsultas.

Tiempo aproximado: **35 minutos**



ACTIVIDAD

Práctica SQL

Cada una de las consignas a continuación corresponde a la consulta de una tabla dentro de la base de datos. Para resolverlas deberás identificar cuándo es necesario que la consulta contenga alguno o varios de los siguientes componentes:

- ✓ Where
- ✓ Operadores lógicos/relacionales.
- ✓ Función de agregación
- ✓ Función escalar.
- ✓ Subconsulta.



ACTIVIDAD

Práctica SQL

1. Indicar cuántos cursos y carreras tiene el área de Data. Renombrar la nueva columna como `cant_asignaturas`. *Keywords: Tipo, Área, Asignaturas.*
2. Se requiere saber cual es el nombre, el documento y el teléfono de los estudiantes que son profesionales en agronomía y que nacieron entre el año 1970 y el año 2000. *Keywords: Estudiantes, Profesión, fecha de Nacimiento.*
3. Se requiere un listado de los docentes que ingresaron en el año 2021 y concatenar los campos nombre y apellido. El resultado debe utilizar un separador: guión (-). Ejemplo: Elba-Jimenez. Renombrar la nueva columna como `Nombres_Apellidos`. Los resultados de la nueva columna deben estar en mayúsculas. *Keywords: Staff, Fecha Ingreso, Nombre, Apellido.*



ACTIVIDAD

Práctica SQL

4. Indicar la cantidad de encargados de docentes y de tutores. Renombrar la columna como CantEncargados. Quitar la palabra "Encargado "en cada uno de los registros. Renombrar la columna como NuevoTipo. Keywords: Encargado, tipo, Encargado_ID.
5. Indicar cuál es el precio promedio de las carreras y los cursos por jornada. Renombrar la nueva columna como Promedio. Ordenar los promedios de Mayor a menor Keywords: Tipo, Jornada, Asignaturas.
6. Se requiere calcular la edad de los estudiantes en una nueva columna. Renombrar a la nueva columna Edad. Filtrar solo los que son mayores de 18 años. Ordenar de Menor a Mayor Keywords: Fecha de Nacimiento, Estudiantes.
7. Se requiere saber el Nombre,el correo, la camada y la fecha de ingreso de personas del staff que contienen correo .edu y su DocentelD se mayor o igual que 100 Keywords: Staff, correo, DocentesID



ACTIVIDAD

Práctica SQL

8. Se requiere conocer el documento, el domicilio, el código postal y el nombre de los primeros 5 estudiantes que se registraron en la plataforma. Keywords: Documento, Estudiantes, Fecha Ingreso.



Definición Modelo relacional



ENTREGABLE N°2

Definición Modelo relacional

Deberán entregar...

Documentación en PDF que incluya:

- ✓ Objetivo del proyecto.
- ✓ Alcance.
- ✓ Usuario final y nivel de aplicación del análisis (Operativo, Táctico, Estratégico)
- ✓ Diagrama de entidad relación de las tablas que contienen la información a analizar.
- ✓ Listado de las tablas. Cada tabla debe tener una descripción de la misma, y también la definición de cada llave primaria y foránea, según corresponda. Así como también describir la relación entre tablas.
- ✓ Listado de las columnas de cada tabla y la información correspondiente, además las PK en todas las tablas y FK dependiendo de las relaciones entre tablas.



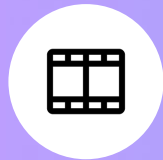
ENTREGABLE N°2

Definición Modelo relacional

Formato:

- ✓ La documentación debe ser entregada en formato pdf y debe llevar por título el nombre del proyecto y el nombre de su autor.

¿Preguntas sobre el 2°
entregable? 🚀



¿Quieres saber más?
**Te dejamos material
ampliado de la clase**



MATERIAL AMPLIADO

Recursos multimedia

- ✓ [7 razones para aprender SQL](#) | CampusMPV.es
- ✓ [5 Bases de datos para la empresa](#) | Francisco Palazón
- ✓ [w3schools](#) | Practicar SQL
- ✓ [Instructivo de subconsultas](#) | Canal en YouTube: Código Facilito
- ✓ [Business Analytics Translator: el rol de la creatividad analítica en la toma de decisiones estratégicas](#) | Josefina Zbar

Muchas gracias.

Resumen de la clase hoy

- ✓ Sintaxis SQL y sus características
- ✓ Operadores SQL
- ✓ Sentencias más comunes SQL

La próxima semana

Los próximos temas que vamos a ver



On demand

- ✓ Video 5.1 – Conociendo la sentencia JOIN
- ✓ Video 5.2 – Inner join
- ✓ Video 5.3 – Left join
- ✓ Video 5.4 – Right join
- ✓ Video 5.5 – Outer join
- ✓ Video 5.6 – Explorando la sentencia UNION
- ✓ Podcast 5.7 – Entrevista a Speaker



Clase en vivo (2 hs)

- ✓ Sentencias JOIN y UNION

Opina y valora
esta clase

¿Sabías que
premiamos a nuestros estudiantes
por su dedicación?

Conoce los [beneficios](#) del Top 10

#DemocratizandoLaEducación