



Clase 05. SQL

CONSULTAS Y SUBCONSULTAS

***RECUERDA PONER A GRABAR LA
CLASE***





OBJETIVOS DE LA CLASE

- Unir dos tablas
- Repasar los tipos de datos
- Profundizar el uso de LIKE
- Reconocer una subconsulta y sus diferentes tipos
- Implementar diferentes subconsultas SQL

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 5

¡Para
recordar!



CONSULTAS SQL

UNIÓN

TIPOS DE DATOS

SELECT * FROM tabla
WHERE (SELECT...)

USO COMBINADO DE
LIKE Y COMODINES

EXPRESIONES
REGULARES

SUBCONSULTAS SQL

SELECT * FROM tabla
WHERE (SELECT...)

COMBINACIÓN DE
SUBCONSULTAS Y
FUNCIONES

(=, AVG, SUM)

CODER HOUSE

UNIÓN

UNIFICAR DOS O MÁS CONSULTAS SELECT

UNION: DEFINICIÓN

El operador de la UNION **combina los resultados de dos o más consultas en un único resultado** que incluye todas las filas que pertenecen a todas las consultas que aparecen

Es decir, las consultas se ejecutan por separado, concatenando luego los resultados de cada una

UNION: DEFINICIÓN

Hay una condición para poder utilizar el operador UNION, y es que la cantidad de columnas en cada consulta o tablas debe ser la misma y del mismo tipo de datos (unión compatible)

GAME (level 1)				
id_game	name	description	id_level	id_class
37	Pokémon Diamante	imperdiet et commodo	1	203
54	Kena: Bridge of Spirits	pellentesque at	1	101
62	Knights of the Old Republic	ipsum primis in faucibus	1	197
77	Dragon Ball Z Kakarot	lorem	1	174
87	First Class Trouble	ipsum primis in faucibus	1	25
99	Chernobylite	fermentum donec ut mauris	1	217

GAME (level 2)				
id_game	name	description	id_level	id_class
1	Forza Horizon 5	odio donec	2	143
5	Age of Empires IV		2	50
8	Blue Reflection: Second Light	libero rutrum ac lobortis	2	18
10	Voice of Cards	parturient montes nascetur	2	275
22	Poppy Playtime	nulla elit ac nulla	2	165
35	Resident Evil 4 VR	magna	2	275
56	PES 2022	mi pede	2	276
66	Ghost Recon: Frontline	habitasse	2	286

EJEMPLO DE UNION

UNION: EJEMPLO

Veamos a continuación cómo combinar dos tablas, o consultas, de igual estructura utilizando la cláusula **UNION**.

GAME (level 1)				
id_game	name	description	id_level	id_class
37	Pokémon Diamante	imperdiet et commodo	1	203
54	Kena: Bridge of Spirits	pellentesque at	1	101
62	Knights of the Old Republic	ipsum primis in faucibus	1	197
77	Dragon Ball Z Kakarot	lorem	1	174
87	First Class Trouble	ipsum primis in faucibus	1	25
99	Chernobylite	fermentum donec ut mauris	1	217

GAME (level 2)				
id_game	name	description	id_level	id_class
1	Forza Horizon 5	odio donec	2	143
5	Age of Empires IV		2	50
8	Blue Reflection: Second Light	libero rutrum ac lobortis	2	18
10	Voice of Cards	parturient montes nascetur	2	275
22	Poppy Playtime	nulla elit ac nulla	2	165
35	Resident Evil 4 VR	magna	2	275
56	PES 2022	mi pede	2	276
66	Ghost Recon: Frontline	habitasse	2	286

```
SELECT id_game, name,  
description, id_level, id_class
```

```
FROM game
```

```
WHERE id_level = 1
```

```
UNION
```

```
SELECT id_game, name,  
description, id_level, id_class
```

```
FROM game
```

```
WHERE id_level = 2;
```

UNION- EJEMPLO

RESULTADO				
id_game	name	description	id_level	id_class
37	Pokémon Diamante	imperdiet et commodo	1	203
54	Kena: Bridge of Spirits	pellentesque at	1	101
62	Knights of the Old Republic	ipsum primis in faucibus	1	197
77	Dragon Ball Z Kakarot	lorem	1	174
87	First Class Trouble	ipsum primis in faucibus	1	25
99	Chernobylite	fermentum donec ut mauris	1	217
1	Forza Horizon 5	odio donec	2	143
5	Age of Empires IV		2	50
8	Blue Reflection: Second Light	libero rutrum ac lobortis	2	18
10	Voice of Cards	parturient montes nascetur	2	275
22	Poppy Playtime	nulla elit ac nulla	2	165
35	Resident Evil 4 VR	magna	2	275
56	PES 2022	mi pede	2	276
66	Ghost Recon: Frontline	habitasse	2	286
90	Ghost of Tsushima	purus	2	285

TIPOS DE DATOS

TIPOS DE DATOS EN LOS CAMPOS

DEFINICIÓN DE LOS DATOS

La clase pasada vimos cómo realizar consultas con **WHERE**, especificando valores como filtros, según el tipo de datos almacenado. Con esto notamos que, en SQL, el **tipo de dato a definir en un campo es un punto clave** debido a diferentes factores.

Por ello, al gestionar nuestras DB debemos establecer reglas de contenidos claras para cada uno de los campos en las tablas.



*Hagamos un rápido repaso de los tipos de datos más
comunes en SQL...*



PRINCIPALES TIPOS DE DATOS

Existen muchos otros tipos de datos en SQL pero, por el momento, solo nos manejaremos con los básicos.

Éstos nos acompañarán durante todo nuestro período de trabajo.

Tipo de dato	valor SQL	Ejemplo
Número entero	int	1000
Texto	text(n)	Coderhouse
Alfanumérico	varchar(n)	AB123CD
Fecha	date	21/03/1975
Fecha y Hora	datetime	21/01/1972 15:00:00
Verdadero o Falso	boolean	TRUE ó FALSE
Decimal	decimal(p, s)	3008,05
Numérico	numeric(p, s)	1407,96

LIKE '0%'

USO DEL OPERADOR LIKE

EL OPERADOR LIKE

La implementación de este operador **se realiza sobre campos del tipo texto o alfanuméricos**, para buscar parte de un valor coincidente.

En combinación con el dato a buscar, **se suele utilizar** al menos un carácter **'comodín'**, que oficia de parámetro para encontrar datos 'que se asimilen a lo escrito'

EJEMPLO DE USO OPERADOR LIKE

Sobre nuestra tabla **GAME** debemos traer aquellos registros cuyo nombre del juego comience con FIFA

```
SELECT *  
FROM game  
WHERE name LIKE 'FIFA%';
```

GAME				
id_game	name	description	id_level	id_class
49	FIFA 22: Ultimate Te	habitasse	14	13
45	FIFA 22	tempor convallis nulla ne	106	8
12	FIFA 22: Ultimate Te	suspendisse	205	14

¿Qué resultados devolverá?👁️👁️

EJEMPLO DE USO OPERADOR LIKE

También podemos combinar el uso de % para obtener parámetros que coincidan con un texto ubicado en cualquier parte del texto almacenado.

```
SELECT *  
FROM game  
WHERE name LIKE '%Ultimate%';
```

GAME				
id_game	name	description	id_level	id_class
49	FIFA 22: Ultimate Te	habitasse	14	13
12	FIFA 22: Ultimate Te	suspendisse	205	14

EJEMPLO DE USO OPERADOR LIKE

Y, por supuesto, podemos buscar también algo más específico, como ser todos aquellos registros que finalicen con el texto en cuestión sin importar lo que tengan al inicio del mismo.

```
SELECT *  
FROM game  
WHERE name LIKE '%Team';
```

GAME				
id_game	name	description	id_level	id_class
49	FIFA 22: Ultimate Team	habitasse	14	13
12	FIFA 22: Ultimate Team	suspendisse	205	14

USO DEL LIKE CON EL COMODÍN “_”

E implementando el caracter comodín “_”, podemos también definir el desconocimiento de un solo caracter.

Se puede combinar con el caracter %.

```
SELECT *  
FROM game  
WHERE name LIKE ‘_IFA%’;
```

GAME				
id_game	name	description	id_level	id_class
49	FIFA 22: Ultimate Te	habitasse	14	13
45	FIFA 22	tempor convallis nulla ne	106	8
12	FIFA 22: Ultimate Te	suspendisse	205	14

COMBINAR LIKE Y CORCHETES []

USO DE LIKE CON CORCHETES []

Dentro de expresiones regulares, el uso de corchetes **nos permite que el resultado de la búsqueda se limite a un rango inicial determinado de caracteres.**

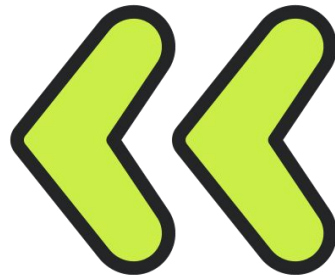
```
SELECT *  
FROM game  
WHERE name LIKE '[A-B]%' ;
```

GAME				
id_game	name	description	id_level	id_class
97	Actraiser Renaissance	viverra eget congue	231	3
5	Age of Empires IV		50	2
46	Alan Wake Remastered	dui luctus rutrum nulla te	137	15
30	Assassins Creed Valhalla	vivamus	207	9
67	Aven Colony	sollicitudin ut	238	14
74	Back 4 Blood	pulvinar	36	9
48	Back 4 Blood	ornare imperdiet sapien	198	4
59	Battlefield 2042	pede libero	31	12
51	Battlefield 2042		153	6
93	Blast Brigade	in sapien iaculis congue	19	8
8	Blue Reflection: Second Lig	libero rutrum ac lobortis	18	2
91	Bright Memory: Infinite		77	5

USO DE LIKE CON CORCHETES []

También podemos integrar la búsqueda de registros que coincidan con uno o dos caracteres iniciales específicos 📍

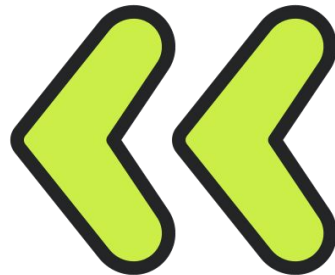
```
SELECT *  
FROM system_user  
WHERE first_name LIKE '[AM]%de1%'
```

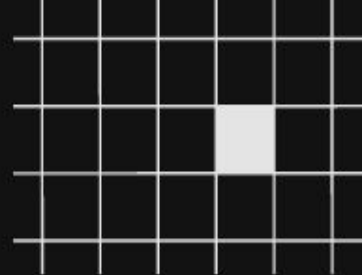


EXCLUSIÓN DE CARACTERES

Aprovechando el uso de corchetes, también nos podemos ocupar de excluir de los resultados determinados caracteres. Para ello debemos usar **[^]**

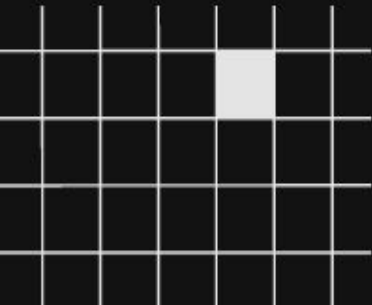
```
SELECT *  
FROM game  
WHERE name LIKE '[^DV]%' ;
```





EJEMPLO EN VIVO

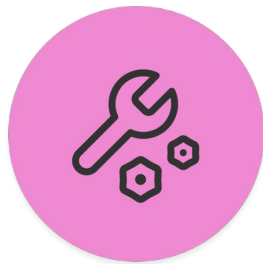
Veamos algunos ejemplos en vivo del uso del operador LIKE





BREAK

¡5/10 MINUTOS Y VOLVEMOS!



USO DEL OPERADOR LIKE

Implementemos el uso del operador **LIKE** sobre las tablas **GAME** y **SYSTEM_USER** de nuestra base de datos **GAMER**

Tiempo estimado: 15 minutos



USO DEL OPERADOR LIKE

Buscaremos usuarios (SYSTEM_USER) utilizando el operador LIKE, y combinando el mismo con las diferentes variantes vistas hasta aquí.

- Aquellos usuarios cuyo nombre comience con la letra 'J'
- Aquellos usuarios cuyo apellido contenga la letra 'W'
- Aquellos usuarios cuyo nombre contenga la letra 'i' en segundo lugar
- Aquellos usuarios cuyo nombre finalice con la letra 'k'
- Aquellos usuarios cuyo nombre no incluya las letras 'ch'
- Aquellos usuarios cuyo nombre solo incluya las letras 'ch'

SUBCONSULTAS SQL

OBTENER DATOS DE OTRAS TABLAS MEDIANTE LAS SUBCONSULTAS



USO DE CONSULTAS DENTRO DE CONSULTAS

Si debemos incluir en la cláusula **WHERE** algún criterio de selección que existe en otra tabla, las subconsultas son el elemento ideal que nos permitirá recuperar los valores acordes a dicha condición.

REGLAS A TENER EN CUENTA

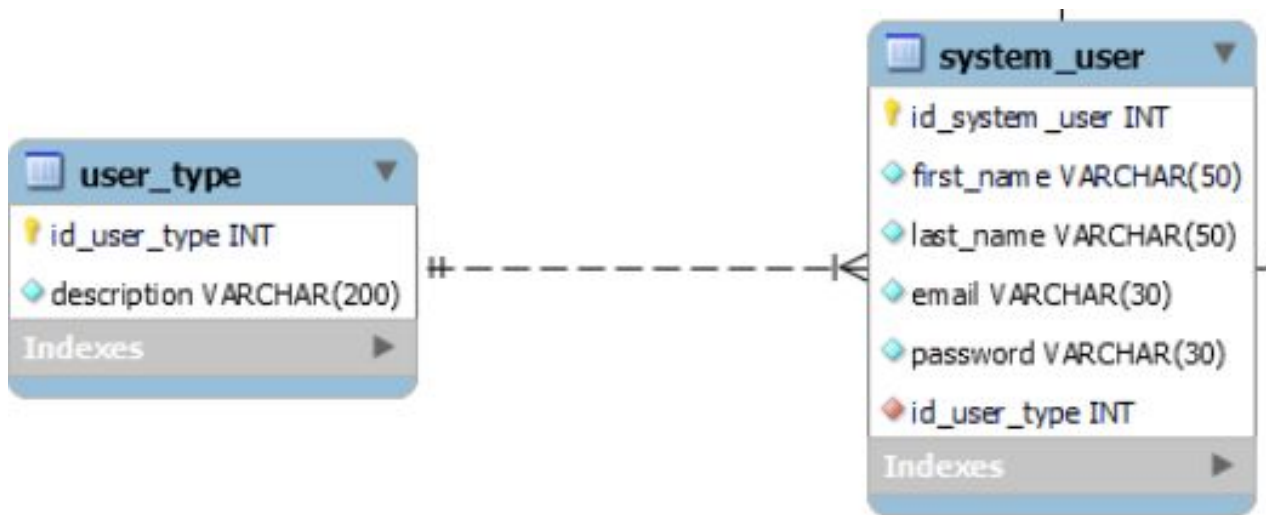
Para poder llevar a cabo esto de manera exitosa, debemos tener en cuenta las siguientes reglas:

- La subconsulta debe ir entre paréntesis.
- La subconsulta debe tener una sola columna o expresión.
- No podemos utilizar **BETWEEN** o **LIKE** en la subconsulta.
- No debemos colocar la cláusula **ORDER BY** en la subconsulta.
- Otras cuestiones más con **UPDATE** y **DELETE**, que veremos oportunamente cuando abordemos dichos temas.

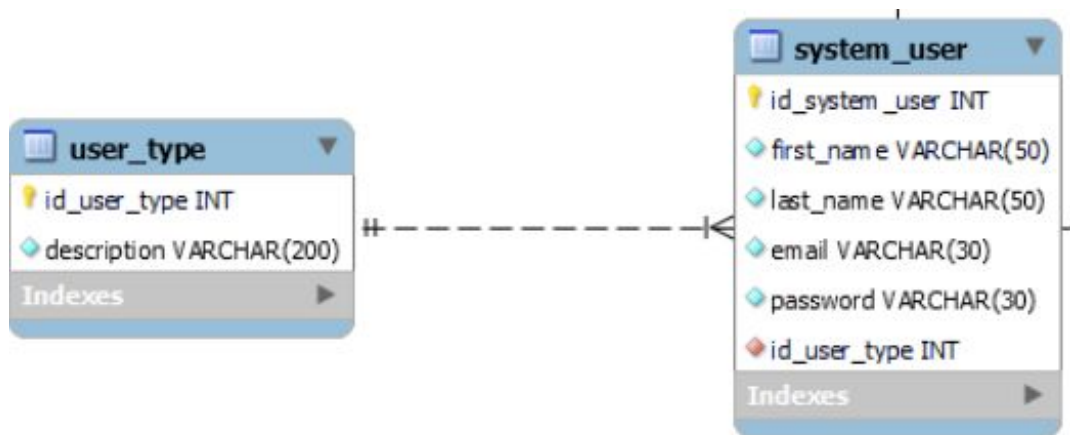
EJEMPLO DE SUBCONSULTAS



Contamos con dos tablas: **SYSTEM_USER** y **USER_TYPE**; necesitamos visualizar aquellos usuarios con el máximo identificador de tipo 🙌



EJEMPLO DE SUBCONSULTAS



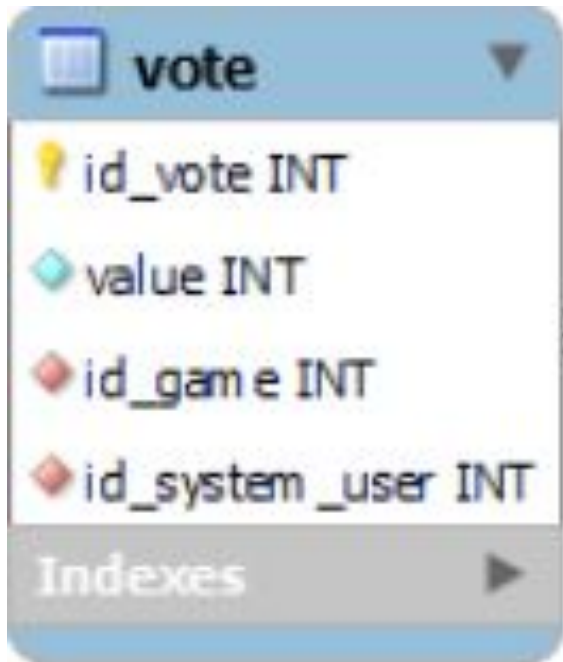
```
SELECT id_system_user, last_name
FROM system_user
WHERE id_user_type = (SELECT max(id_user_type) FROM user_type);
```

SUBCONSULTAS EN UNA MISMA TABLA

SUBCONSULTAS INTERNAS

También podemos abordar subconsultas dentro de una misma tabla.

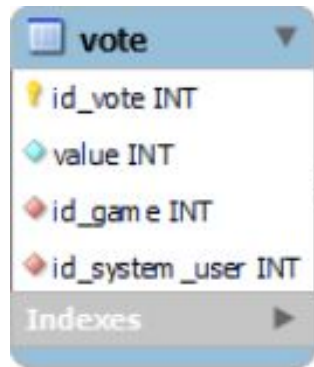
En este caso, la **tabla vote** cuenta con información del puntaje que cada usuario le dio a un juego en la **columna value**.



SUBCONSULTAS INTERNAS

Busquemos los usuarios que votaron con un puntaje superior al promedio.

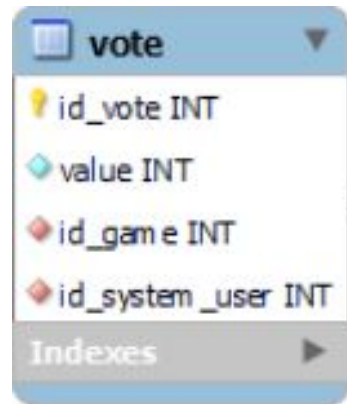
Nota: La función floor convierte float a entero.



```
SELECT id_system_user
FROM vote WHERE value = (SELECT FLOOR(AVG(value)) FROM vote);
```

SUBCONSULTAS INTERNAS

Podemos también obtener los votos totales de un juego específico, por ejemplo el de menor **id**:



```
SELECT SUM(value)
FROM vote
WHERE id_game = (SELECT min(id_game) FROM game);
```

SUBCONSULTAS INTERNAS

Por supuesto, podemos también sumar otros operadores combinados con **WHERE**, como ser **<**, **>**, **=**, **<=**, para recuperar un conjunto de valores más específico todavía.

Por ejemplo los usuarios que votaron por encima del promedio total de votos.

```
SELECT id_system_user  
FROM vote  
WHERE value > (SELECT avg(value) FROM vote);
```

ORDENAMIENTO DE SUBCONSULTAS SQL

USO DE ORDER BY EN SUBCONSULTAS SQL

ORDENAMIENTO DE CONSULTAS CON SUBCONSULTAS

Sabemos que el ordenamiento de la información obtenida a partir de una consulta es clave para mostrar los resultados de forma homogénea.

Por ello, la **sentencia ORDER BY** puede ser utilizada dentro de consultas con subconsultas, teniendo en cuenta que dicho ordenamiento debe realizarse en la consulta principal.

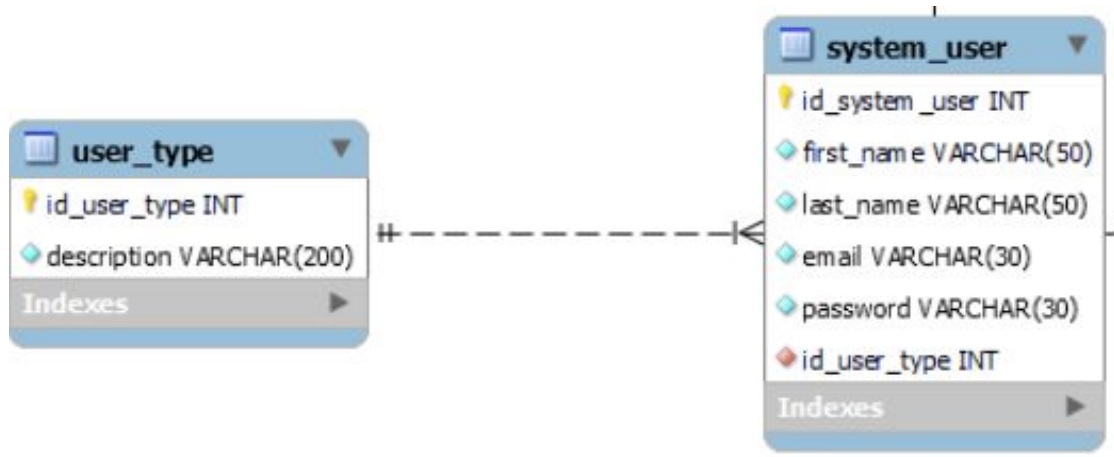
ORDENAMIENTO DE CONSULTAS CON SUBCONSULTAS

Recordemos que en los casos vistos hasta ahora siempre se ejecuta la **subconsulta en primera instancia** y una vez resuelta ésta, se ejecuta la consulta.

Entonces debemos incluir el orden sobre la consulta más externa.



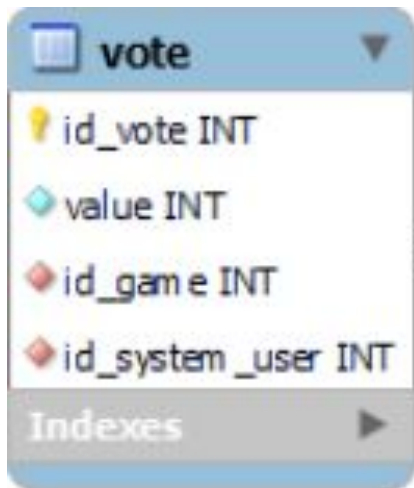
EJEMPLOS DE SUBCONSULTAS



```
SELECT id_system_user, last_name
FROM system_user
WHERE id_user_type = (SELECT max(id_user_type) FROM user_type)
ORDER BY last_name ASC;
```


USO DE GROUP BY EN SUBCONSULTAS SQL

INTEGRANDO GROUP BY EN CONSULTAS Y SUBCONSULTAS



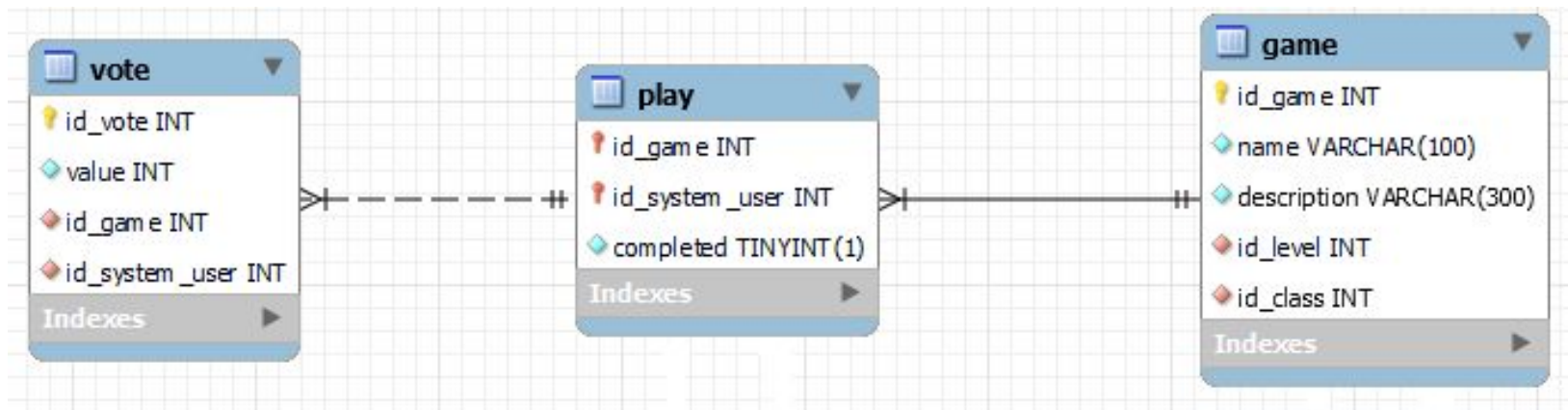
Acoplamos la sentencia **GROUP BY** dentro de una **consulta** con **subconsulta asociada**.

Veamos que nuestro ejemplo presta más información que antes, destacando ahora en qué se desempeña cada uno de ellos

INTEGRANDO GROUP BY EN CONSULTAS Y SUBCONSULTAS

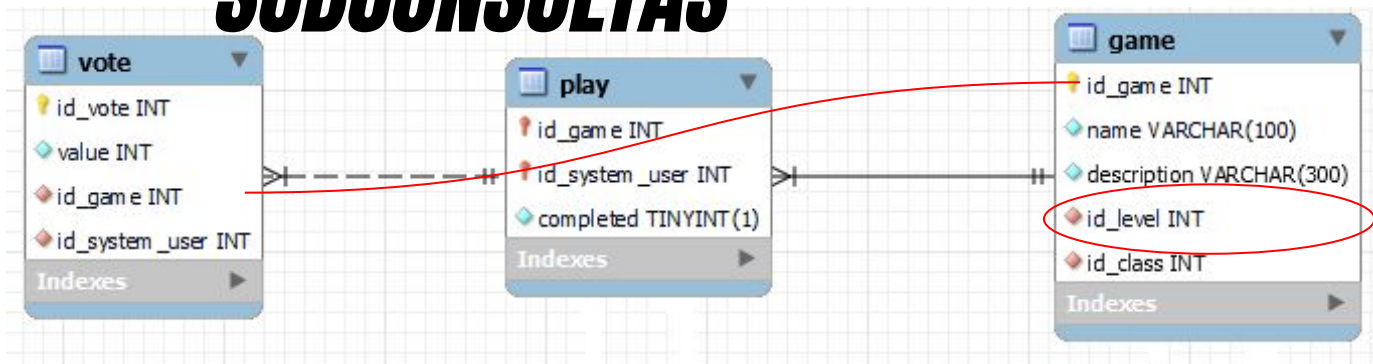
Debemos obtener la suma de votos por juego, solo de aquellos juegos de nivel 1.

Observemos cómo podemos hacerlo con nuestras tablas



INTEGRANDO GROUP BY EN CONSULTAS Y SUBCONSULTAS

Obtener la suma de votos **por juego**, solo de aquellos juegos de nivel 1.



```
SELECT id_game, SUM(value) AS votos
FROM vote
WHERE id_game IN (SELECT id_game
                  FROM game WHERE id_level = 1)
GROUP BY id_game;
```

INTEGRANDO GROUP BY EN CONSULTAS Y SUBCONSULTAS

De esta forma, podremos conocer cuál es el juego que más puntos acumula con votos, o el que menos si ordenamos el resultado.

La flexibilidad de SQL nos permite realizar cualquier consulta, siempre que los diseños estén correctamente normalizados.

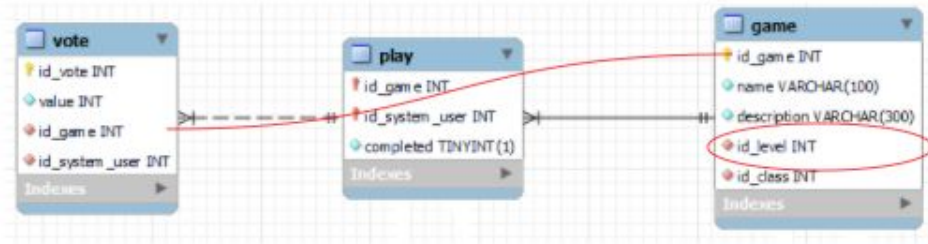
USO DE HAVING EN SUBCONSULTAS SQL

INTEGRANDO HAVING EN CONSULTAS Y SUBCONSULTAS

Veamos cómo integrar consultas con **HAVING**. Ahora seleccionaremos los juegos pero sólo aquellos que hayan tenido más de un voto.

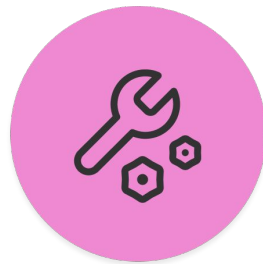


INTEGRANDO HAVING EN CONSULTAS Y SUBCONSULTAS



Seleccionaremos los juegos
pero sólo aquellos que hayan
tenido más de un voto 🙌

```
SELECT id_game, name
FROM game
WHERE id_level = 1 AND
      id_game IN
      (SELECT id_game
       FROM vote
       GROUP BY id_game
       HAVING count(*) > 1);
```

SUBCONSULTAS SQL

Con base en la estructura previamente generada, resolver las consignas presentadas.

Tiempo estimado: 10 minutos

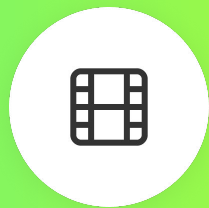
SUBCONSULTAS SQL

Desafío
generico



Llevemos todos los ejemplos hasta aquí aprendidos, a la base de datos **GAMER**. Trabajamos con las tablas combinando consultas y subconsultas que cumplan con el uso de:

- Juegos jugados por jugador
- Condicionales en el nombre de los usuarios
- Integración de HAVING
- Funciones de agregación y GROUP BY



***¿QUIERES SABER MÁS? TE DEJAMOS
MATERIAL AMPLIADO DE LA CLASE***



- Subconsultas SQL: <https://youtu.be/rGPb5E1UAJA> | Píldoras informáticas
- Subconsultas en SQL SERVER: <https://youtu.be/cQ5SwUhBQI>

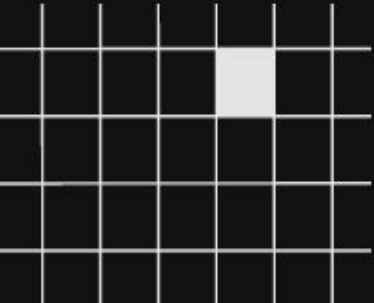
¿PREGUNTAS?





¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- Uso de UNION para unificar tablas.
 - Uso de tipos de datos, LIKE y comodines.
 - Subconsultas SQL y sus diferentes tipos.
 - Implementar subconsultas SQL.
- 



OPINA Y VALORA ESTA CLASE

#DEMOCRATIZANDO LA EDUCACIÓN

CODER HOUSE