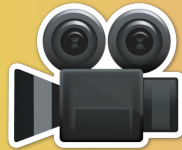




Clase 09. SQL

VISTAS

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





OBJETIVOS DE LA CLASE

- Crear, actualizar y eliminar vistas.
- Identificar ventajas y desventajas de cada tipo de vista

CRONOGRAMA DEL CURSO

Clase 8



Manejo de Tablas



TIPOS DE TABLAS (HECHO, TRANSACCIONALES, DIMENSIONALES)



MODIFICAR ESTRUCTURA DE TABLAS



DESCRIPCIÓN DE TABLAS

Clase 9



Vistas



TIPOS DE VISTAS, IMPLEMENTACIÓN Y USO



PRÁCTICAS CON VISTAS

Clase 10



Workshop I



REPASO DE LO APRENDIDO HASTA AHORA

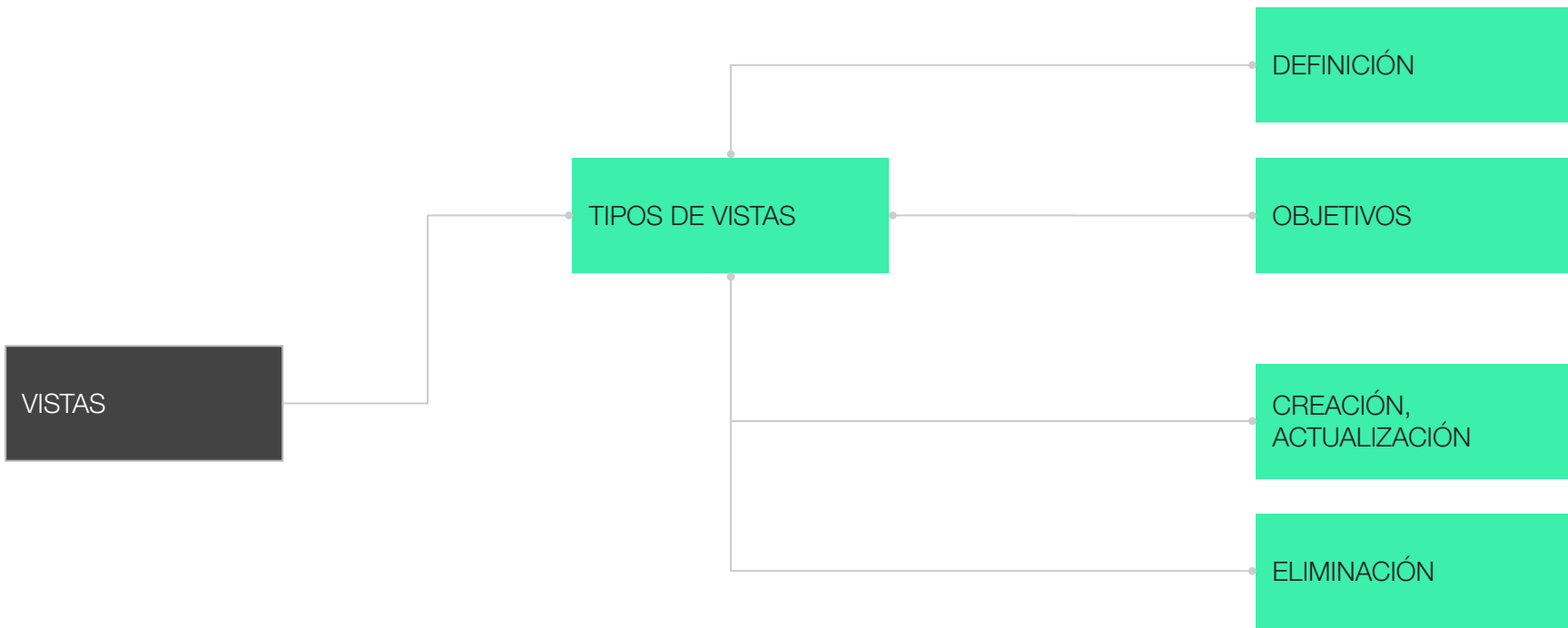


ENTREGA INTERMEDIA

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 9

¡Para
recordar!



VISTAS, DEFINICIÓN Y OBJETOS

CONCEPTO GENERAL

DEFINICIÓN DE VISTAS SQL

Tal como mencionamos algunas clases atrás, una **Vista SQL** es básicamente una **tabla virtual que se genera a partir de la ejecución de una o más consultas SQL**, aplicada sobre una o más tablas.



VISTAS: definición

Su estructura corresponde a una **serie de filas y**

columnas tal como encontramos en las tablas

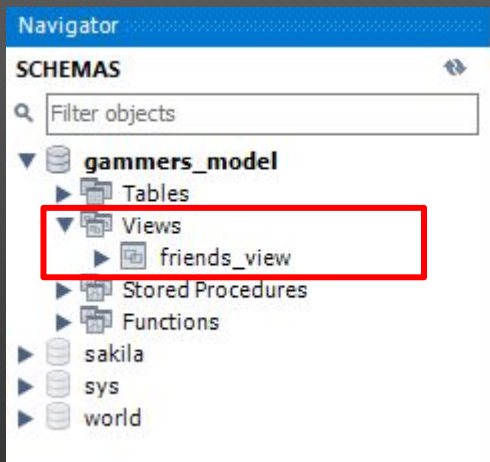
SQL, que **almacenan la vista de la información**

tal como la definimos al crearla.

VISTAS: almacenamiento

Podemos guardar las Vistas con un nombre distintivo, para poder ejecutarlas cuantas veces consideremos necesario.

Su almacenamiento se da en la sección **Views** de los objetos de la base de datos.



VISTAS: sintaxis



Su sintaxis está compuesta por:

- La sentencia CREATE VIEW.
- El nombre que deseamos darle a la Vista.
- Las columnas que se crearán.
- La consulta SQL desde donde obtendrá los datos.

```
CREATE VIEW nombre_vista  
[lista_columnas]  
AS consulta_sql
```

VISTAS: sintaxis extendida



La sintaxis **CREATE VIEW** cuenta con un modificador (*opcional*) denominado **OR REPLACE**.

Esta sintaxis se ocupa de, crear la Vista si no existe, o reemplazar la existente por una nueva.

```
CREATE VIEW OR REPLACE nombre_vista  
[lista_columnas]  
AS consulta_sql
```

BENEFICIOS

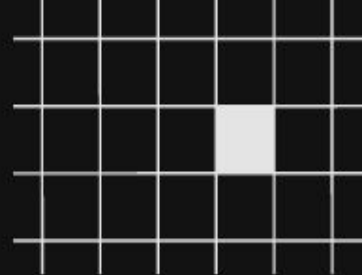
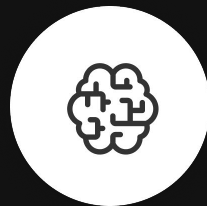
VISTAS: beneficios

- **Privacidad de la información:** los usuarios podrán ver sólo aquellos datos que creamos convenientes mostrar, en otras palabras, mejora la seguridad de la DB.
- **Rendimiento de la DB:** Crear queries sobre vistas complejas nos ahorra ejecutar una query pesada para llegar a la información.
- **Protección de datos:** Aquellos usuarios que no poseen un entorno de pre-producción, las vistas evitan errores de borrado o alteración.

Vistas: protección de datos



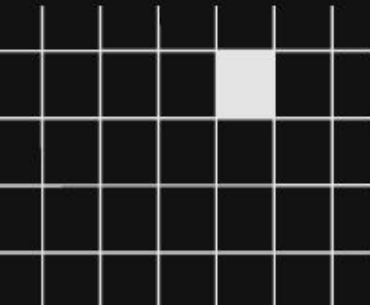
Construir vistas facilitan el acceso a mucha información además de ser la opción más práctica y segura para proteger la información de las tablas.



¡PARA PENSAR!

Las vistas, a pesar de ser objetos que toman y muestran información de una tabla, tienen el poder de insertar nuevos registros en esta.

¿VERDADERO O FALSO?
CONTESTA LA ENCUESTA DE ZOOM





EJEMPLO EN VIVO

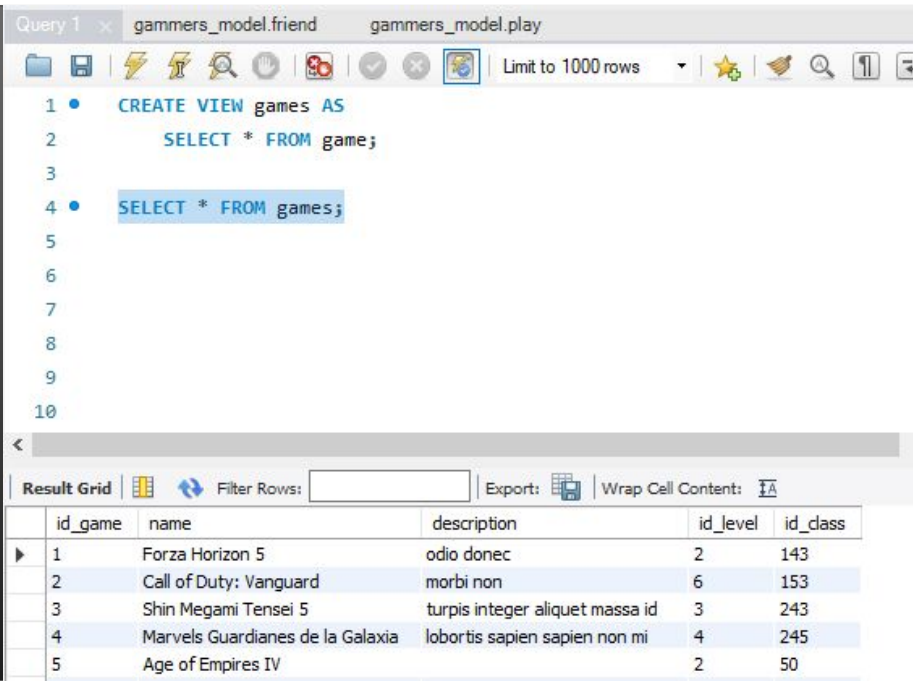
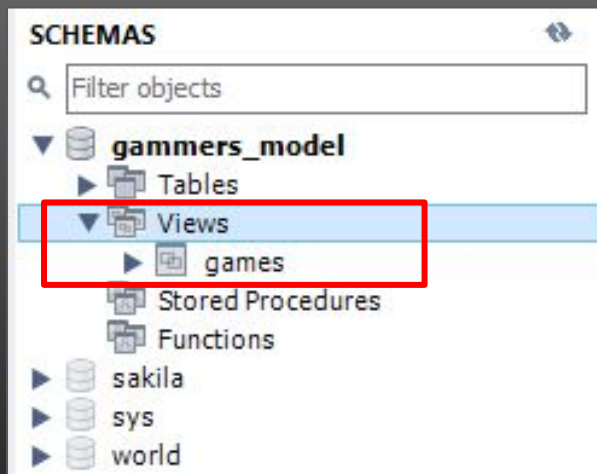
Ahora veremos desde SGBD, cómo crear una vista.

VISTAS: EJEMPLO SIMPLE

Ejemplo
en vivo



Creemos una vista simple denominada **games** de la tabla **game**, trayendo en esta primera instancia, todo su contenido.





CREACIÓN DE VISTAS

Como podemos apreciar, la creación y ejecución de Vistas de datos es muy fácil de realizar.

Los beneficios son varios. Por el momento hemos visto algo básico pero, en breve, comenzaremos a darle una mejor utilidad a esta fabulosa herramienta.



BREAK

¡5/10 MINUTOS Y VOLVEMOS!

TIPOS DE VISTAS

UNA TABLA, MÚLTIPLES RESPUESTAS

Empecemos a sacar provecho de los beneficios de las Vistas, obteniendo múltiples visualizaciones de la información de éstas.



Tipos de Vistas

De una simple tabla podemos aprovechar y obtener múltiples Vistas que nos brindan información de todo tipo y para diferentes públicos.

Veamos, a continuación, cómo sacar más provecho de éstas.

	id_game	name	description	id_level	id_class
▶	1	Forza Horizon 5	odio donec	2	143
	2	Call of Duty: Vanguard	morbi non	6	153
	3	Shin Megami Tensei 5	turpis integer aliquet massa id	3	243
	4	Marvels Guardianes de la Galaxia	lobortis sapien sapien non mi	4	245
	5	Age of Empires IV		2	50

***VISUALIZAR SOLO NOMBRES y
DESCRIPCIONES DE LOS VIDEOJUEGOS***

CODER HOUSE

Nombre y Descripción de los videojuegos

Ejemplo
en vivo



Creamos ahora una vista que se ocupe de mostrarnos los nombres y descripción de los videojuegos

The screenshot shows a database management interface. At the top, there's a toolbar with various icons and a 'Limit to 1000 rows' dropdown. Below the toolbar, the SQL editor contains the following queries:

```
1 • CREATE OR REPLACE VIEW games AS
2     (SELECT name, description FROM game);
3
4 • SELECT * FROM games;
5
6
7
8
9
10
```

Below the SQL editor, there's a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays the following data:

name	description
Forza Horizon 5	odio donec
Call of Duty: Vanguard	morbi non
Shin Megami Tensei 5	turpis integer aliquet massa id
Marvels Guardianes de la Galaxia	lobortis sapien sapien non mi
Age of Empires IV	

CODER HOUSE

***VISUALIZAR SOLO LOS VIDEOJUEGOS CON
NOMBRE CALL OF DUTY***



Video juego Call of Duty

Creamos ahora una vista que se ocupe de mostrarnos los nombres y descripción de los videojuegos que contenga el nombre Call of Duty

```
1 • CREATE OR REPLACE VIEW games AS
2   (SELECT name, description
3     FROM game
4     WHERE name like upper('%Call of Duty%'));
5
6 • SELECT * FROM games;
7
8
9
10
```

Limit to 1000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

name	description
Call of Duty: Vanguard	morbi non
Call of Duty: Vanguard	in felis eu sapien
Call of Duty: Vanguard	proin eu
Call of Duty: Vanguard	odio donec vitae nisi

***CREAR UNA VISTA YA
EXISTENTE***



Crear una Vista ya existente

Intentemos ejecutar un script para crear una Vista nueva con un nombre de Vista que exista previamente.

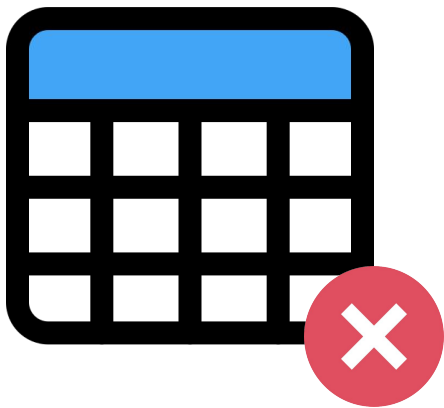
The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, execution, and search. The main area displays a SQL script with line numbers 1 through 8. The script is as follows:

```
1 • CREATE VIEW games AS
2   (SELECT name, description
3     FROM game
4     WHERE name like upper('%Call of Duty%'));
5
6 • SELECT * FROM games;
7
8
```

Below the script, the 'Output' section is visible, showing a table with columns: #, Time, Action, and Message. The first row of the output table is highlighted with a red border and contains an error message:

#	Time	Action	Message
1	07:47:31	CREATE VIEW games AS (SELECT name, description FROM game WHERE name ...	Error Code: 1050. Table 'games' already exists

Crear una Vista ya existente



Nos encontraremos con un error que arrojará MySQL directamente en el panel **Action Output**, alertando que la Vista que deseamos crear, ya existe.

LA SENTENCIA OR REPLACE



CREATE OR REPLACE VIEW

Tal como vimos al inicio de esta clase, para sobrescribir Vistas con una nueva consulta, debemos agregar la sentencia **OR REPLACE**.



```
1 • CREATE OR REPLACE VIEW games AS
2   (SELECT name, description
3     FROM game
4     WHERE name like upper('%Call of Duty%'));
5
6 • SELECT * FROM games;
```

CREATE OR REPLACE VIEW

Ejemplo
en vivo



De esta manera, Mysql sobrescribirá directamente la Vista existente con el código de esta nueva Vista generada. **Tengamos precaución porque Mysql no advertirá la operación de sobre-escritura.**

```
1 CREATE OR REPLACE VIEW games AS
2 (SELECT name, description
3  FROM game
4  WHERE name like upper('%Call of Duty%'));
5
6 SELECT * FROM games;
```

Output

Action Output

#	Time	Action	Message
✓ 1	07:50:14	CREATE OR REPLACE VIEW games AS (SELECT name, description FROM game ...	0 row(s) affected

CODER HOUSE

APLICAR FILTROS A UNA VISTA

APLICAR FILTROS A UNA VISTA

Ejemplo
en vivo



Las Vistas, una vez creadas, pueden ser tratadas tal como si fuesen una tabla, por ejemplo aplicando un orden a la misma.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • CREATE OR REPLACE VIEW games AS
2   (SELECT name, description
3    FROM game
4    ORDER BY name DESC);
5
6 • SELECT * FROM games;
7
```

Below the editor, the 'Result Grid' tab is active, displaying the results of the second query. The results are shown in a table with two columns: 'name' and 'description'.

name	description
Youtubers Life 2	lacus morbi quis
Xbox Game Pass	eleifend pede libero quis
Voice of Cards	parturient montes nascetur
Until You Fall	sollicitudin ut suscipit a feugiat
UnMetal	at turpis
The Last of Us 2	quam turpis adipiscing
The Good Life	ultrices phasellus id sapien in
The Elder Scrolls V: Skyrim Anniversary Edition	proin at

CODER HOUSE

APLICAR FILTROS A UNA VISTA

Ejemplo
en vivo



Si aplicamos una cláusula **WHERE** y/u **ORDER BY**, veremos que los datos de la Vista, se listaran tal como lo indicamos en la consulta de selección.

The screenshot shows a database management tool interface. The top toolbar includes icons for file operations, execution, and search. The SQL editor contains the following code:

```
1 • CREATE OR REPLACE VIEW games AS
2   (SELECT name, description
3    FROM game
4    WHERE name LIKE '%You%'
5    ORDER BY name DESC);
6
7 • SELECT * FROM games;
```

Below the editor is a "Result Grid" section with a "Filter Rows" input field, an "Export" button, and a "Wrap Cell Content" checkbox. The grid displays two rows of data:

name	description
Youtubers Life 2	lacus morbi quis
Until You Fall	sollicitudin ut suscipit a feugiat

Por lo tanto, en las vistas podemos aplicar todas las operaciones que vimos sobre tablas!!

CODER HOUSE

CREAR UNA VISTA CON MÁS DE UNA TABLA

VISTAS CON MÁS DE UNA TABLA

Ejemplo
en vivo



Realicemos una viste que liste los nombres y descripción de los distintos videojuegos que ningún usuario ha podido completar.

VISTAS CON MÁS DE UNA TABLA

Ejemplo
en vivo



Limit to 1000 rows

```
1 • CREATE OR REPLACE VIEW games AS
2   ( SELECT DISTINCT name, description
3     FROM game v JOIN PLAY p ON (v.id_game = p.id_game)
4     where p.completed = false);
5
6 • SELECT * FROM games;
7
```

Result Grid

	name	description
►	Forza Horizon 5	odio donec
	Call of Duty: Vanguard	morbi non
	Shin Megami Tensei 5	turpis integer aliquet massa id
	Football Manager 22	nulla suspendisse potenti
	Football Manager 22	mauris lacinia sapien quis libero
	Blue Reflection: Second Light	libero rutrum ac lobortis
	Darkest Dungeon II	erat nulla
	Voice of Cards	parturient montes nascetur

CODER HOUSE

MÚLTIPLES VISTAS, MÚLTIPLES FUNCIONALIDADES

¡Para
pensar!



Simplificación de tareas mediante las Vistas

Como podemos apreciar, las Vistas nos permiten no sólo llegar a la información concisa mucho más rápidamente, sino que también aportan una variedad de opciones para el trabajo cotidiano sobre los datos de un sistema de software productivo.

Teniendo en cuenta estas dos tablas que integramos recientemente:

¿Qué otras Vistas se te ocurre que podríamos crear utilizando el modelo gamers??

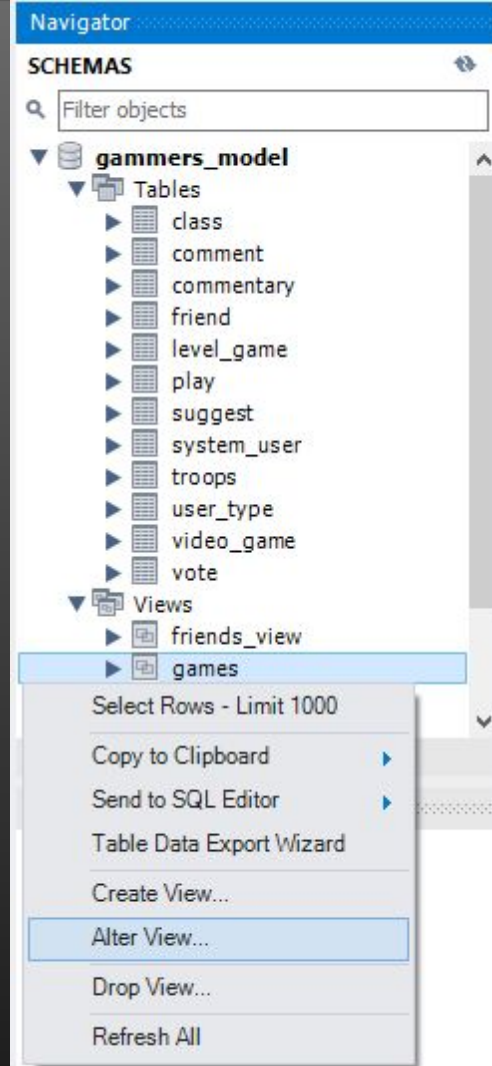
MODIFICAR UNA VISTA EXISTENTE

MODIFICAR UNA VISTA EXISTENTE

Las Vistas ya creadas, pueden ser
modificadas de forma rápida,
recurriendo a

menú contextual > Alter View...

El código de la Vista será editado en una
pestaña de Script.

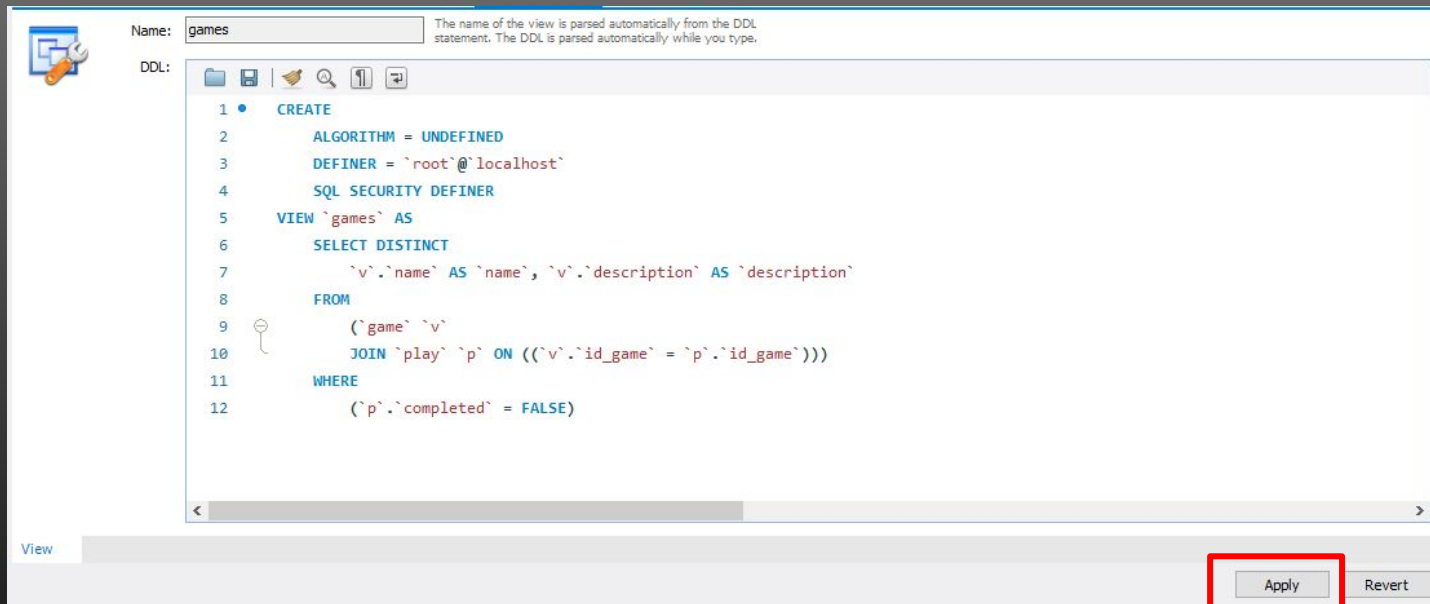


MODIFICAR UNA VISTA EXISTENTE

Ejemplo
en vivo



Aplicando las modificaciones necesarias sobre este código, podremos ejecutarlo a través del botón **Apply**.



MODIFICAR UNA VISTA EXISTENTE

Si vamos a mantener el nombre de la Vista, tal cual existía, recordemos agregar a la sentencia **CREATE VIEW**, el comando **OR REPLACE**, para que los cambios se sobre-escriban correctamente.

ELIMINAR UNA VISTA EXISTENTE

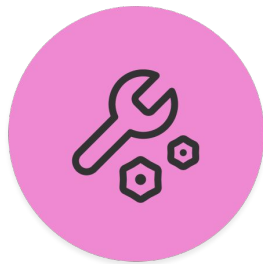
ELIMINAR UNA VISTA

También contamos con la posibilidad de eliminar aquellas Vistas que ya no utilizamos, haciendo uso del comando

DROP VIEW <nombreDeLaVista>.

Ten presente que la eliminación, mediante un script, será instantánea.

```
DROP VIEW vista_productos;
```



PRÁCTICAS CON VISTAS

Desarrollar un scripts SQL e implementar la creación de diferentes tipos de vistas.

Tiempo estimado: 15 minutos

PRÁCTICAS CON VISTAS

Desafío
generico



Sobre el esquema gamers, crea las siguientes Vistas que:

- Muestre first_name y last_name de los usuarios que tengan mail 'webnode.com'
- Muestre todos los datos de los juegos que han finalizado.
- *Muestre los distintos juegos que tuvieron una votacion mayor a 9.*
- *Muestre nombre, apellido y mail de los usuarios que juegan al juego FIFA 22.*

Tiempo estimado: 15 minutos

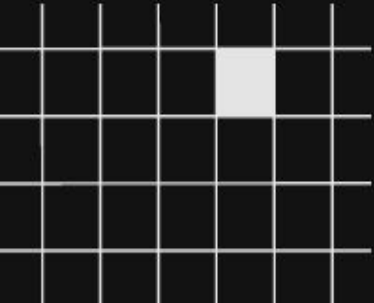
¿PREGUNTAS?





¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- *Vistas, definición y objetivos.*
 - *Ventajas y desventajas de cada tipo de vista.*
- 



OPINA Y VALORA ESTA CLASE

#DEMOCRATIZANDO LA EDUCACIÓN

CODER HOUSE