



Clase 18. Curso SQL

WORKSHOP DML

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





OBJETIVOS DE LA CLASE

- Implementar las vistas creadas.
- Implementar las funciones creadas.
- Implementar los stored procedures creados.
- Implementar los triggers creados.

CRONOGRAMA DEL CURSO

Clase 17



Trigger



CREACIÓN DE TRIGGER



SCRIPT DE CREACIÓN DE TRIGGERS

Clase 18



Workshop DML



COMPARTIENDO PROYECTOS



SEGUNDA ENTREGA DEL PROYECTO FINAL

Clase 19



Sublenguaje DCL



CONCEPTO, SENTENCIAS GRANT Y REVOKE. IMPLEMENTACIÓN DEL SUBLenguaje



PRÁCTICAS SQL

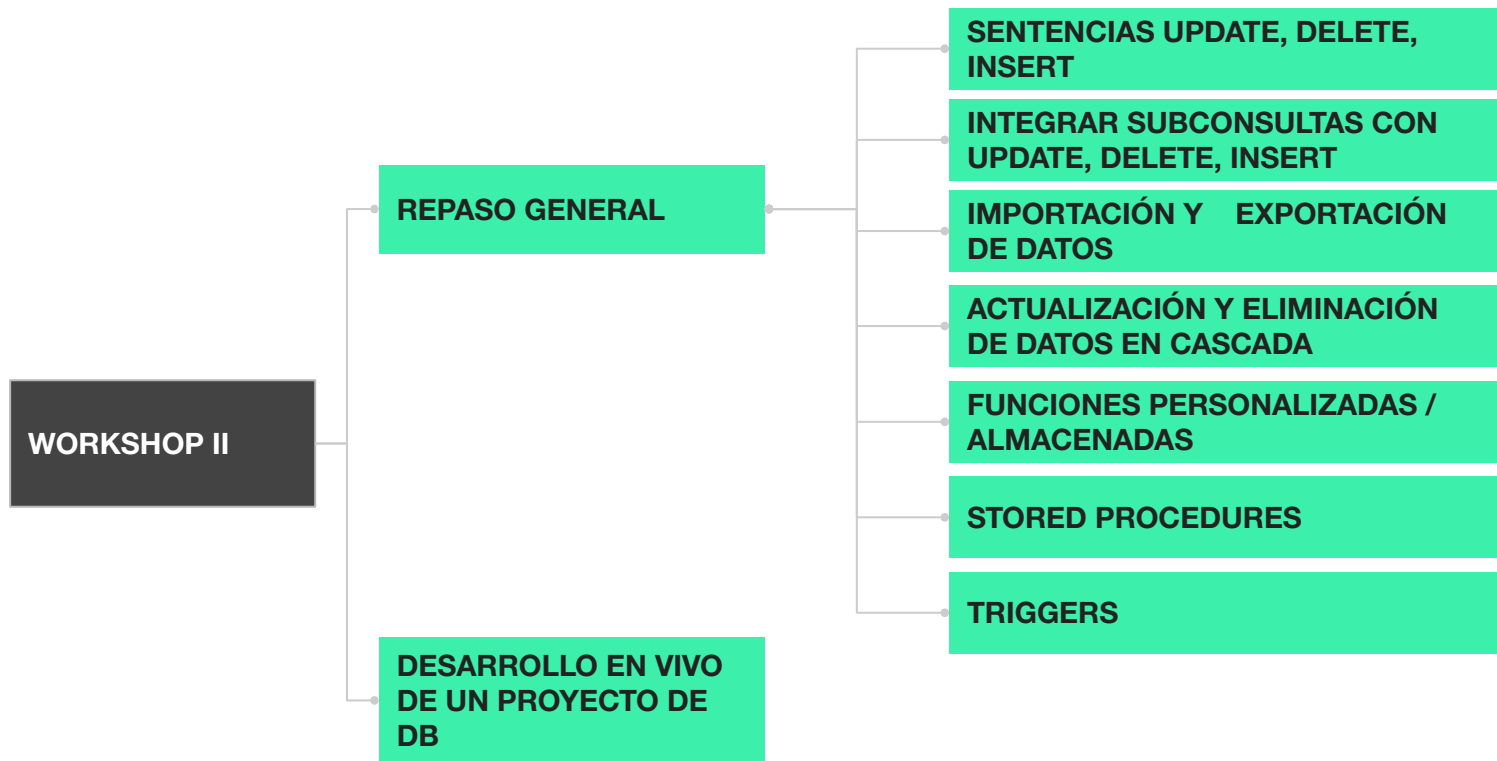


PRESENTAR EN FORMATO SQL LAS SENTENCIAS QUE COMPONEN EL SUBLenguaje DCL

MAPA DE CONCEPTOS

MAPA DE CONCEPTOS CLASE 18

¡Para
recordar!



REPASO GENERAL

REPASO GENERAL

Realicemos un rápido repaso por los conceptos aprendidos brindados en estas últimas 7 clases del lenguaje SQL.

Luego, llevaremos lo aprendido hacia el proyecto final aplicándolo de forma efectiva.



¿TODO LISTO?



CLASE 11 - DML: ACTUALIZACIÓN, MODIFICACIÓN Y ELIMINACIÓN DE DATOS

- Conocimos los conceptos avanzados del lenguaje DML.
- Agregamos registros a las tablas mediante INSERT.
- Modificamos registros en las tablas mediante UPDATE.
- Eliminamos registros de las tablas mediante DELETE.

```
INSERT INTO tabla (campo1, campo2, campo3) VALUES ('valor1', 'valor2', 'valor3');
```

```
UPDATE tabla (campo1, campo2, campo3) SET ('valor1', 'valor2', 'valor3') WHERE =
```

```
...;
```

```
DELETE FROM tabla WHERE campo2 = 'valor2';
```

CLASE 12- SUBCONSULTAS PARA ACTUALIZAR, MODIFICAR Y ELIMINAR DATOS

- Aprendimos el concepto de Subconsultas en DML.
- Utilizamos INSERT combinado con subconsultas.
- Utilizamos UPDATE combinado con subconsultas.
- Utilizamos DELETE combinado con subconsultas.

```
INSERT INTO new_level_game (id_level, description) (  
  
SELECT DISTINCT id_level, 'New level' FROM new_class WHERE id_level NOT IN (  
  
        SELECT id_level FROM level_game));
```

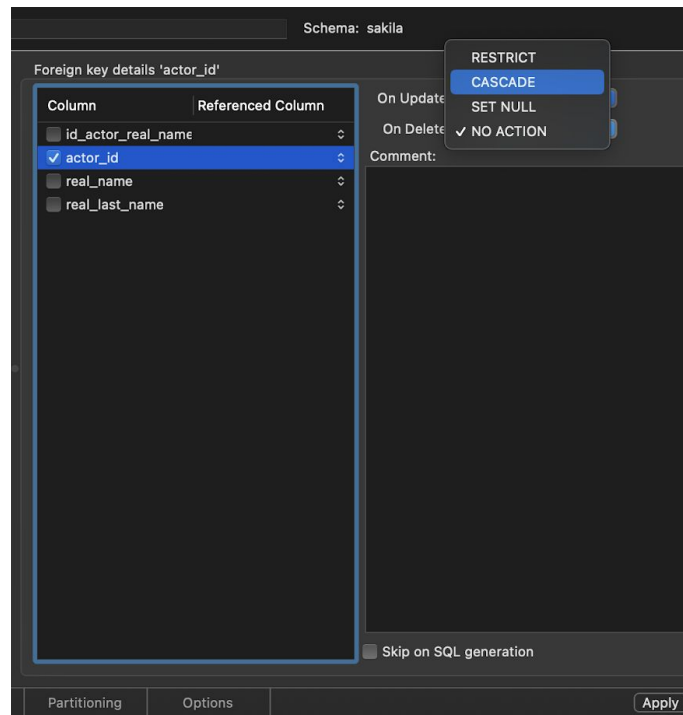
CLASE 13- INSERTAR DATOS MEDIANTE IMPORTACIÓN DE LOS MISMOS

- Conocimos los diferentes modelos de importación de datos.
- Integramos archivos CSV y JSON.
- Aprendimos a optimizar los datos previos a importarlos.
- Importamos datos mediante la línea de comandos Mysql.

```
mysql> load data local infile  
  
    '/Users/Downloads/nuevosproductos.csv'  
  
into table productos  
  
fields terminated by ','  
  
terminated by '\r\n';
```

CLASE 14- ACTUALIZACIÓN Y ELIMINACIÓN DE DATOS EN CASCADA

- Implementamos la actualización y eliminación de datos masivos.
- Aprendimos las Restricciones de Integridad Referencial.
- Integramos el comando DELETE CASCADE y el comando UPDATE CASCADE.



CLASE 15- FUNCIONES ALMACENADAS

- Adquirimos el concepto de una función almacenada y sus ventajas.
- Repasamos su sintaxis y qué sentencias de Transact-SQL pueden integrarse en éstas.
- Aprendimos a armar la estructura de una Función almacenada.
- Creamos nuestra propia función almacenada.

```
1 • CREATE FUNCTION `calcular_litros_de_pintura` (largo INT, alto INT, total_manos INT)
2 RETURNS FLOAT
3 NO SQL
4 BEGIN
5     DECLARE resultado FLOAT;
6     DECLARE litro_x_m2 FLOAT;
7     SET litro_x_m2 = 0.10;
8     SET resultado = ((largo * alto) * total_manos) * litro_x_m2;
9     RETURN resultado;
10 END
```

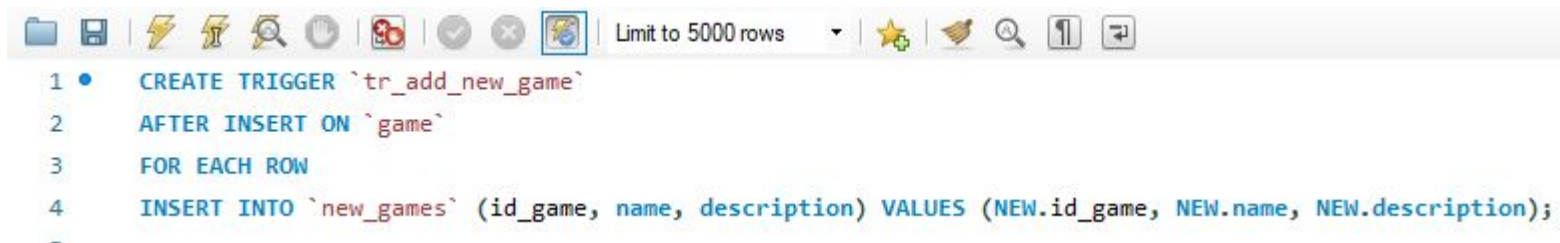
CLASE 16- STORED PROCEDURES

- Aprendimos los Fundamentos de los Stored Procedures y sus beneficios.
- Repasamos su sintaxis y las sentencias de Transact-SQL que pueden utilizarse.
- Aprendimos a armar un Stored Procedure.
- Conocimos las diferencias entre EXEC, EXECUTE y CALL.
- Implementamos nuestros propios Stored Procedures.

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `sp_get_games_order`(IN field CHAR(20))
2 BEGIN
3     IF field <> '' THEN
4         SET @game_order = concat('ORDER BY ', field);
5     ELSE
6         SET @game_order = '';
7     END IF;
8     SET @clausula = concat('SELECT * FROM gammers_model.game ', @game_order);
9     PREPARE runSQL FROM @clausula;
10    EXECUTE runSQL;
11    DEALLOCATE PREPARE runSQL;
12 END
```

CLASE 17- TRIGGERS

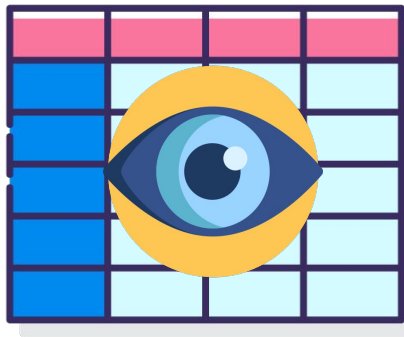
- Aprendimos los Fundamentos de los Triggers y los diferentes tipos de disparadores.
- Repasamos su sintaxis y cómo se estructura un Trigger, de acuerdo a nuestra necesidad.
- Implementamos nuestros propios Triggers bajo diferentes ámbitos donde pueden ser necesarios.



The screenshot shows a SQL IDE interface with a toolbar at the top containing icons for file operations, execution, and search. Below the toolbar, a SQL query is displayed in a text editor. The query is a CREATE TRIGGER statement for a trigger named 'tr_add_new_game' that fires AFTER INSERT ON the 'game' table. The trigger body consists of a FOR EACH ROW loop followed by an INSERT INTO statement that inserts data from the 'new_games' table into the 'game' table. The query is as follows:

```
1 • CREATE TRIGGER `tr_add_new_game`  
2   AFTER INSERT ON `game`  
3   FOR EACH ROW  
4   INSERT INTO `new_games` (id_game, name, description) VALUES (NEW.id_game, NEW.name, NEW.description);  
-
```


ADICIONALES A LO APRENDIDO HASTA AQUÍ



Junto a todo lo aprendido en estas últimas clases, integramos:

- Ejercicios prácticos sobre cada punto abordado.
- Integramos los diferentes conceptos aprendidos en la primera etapa del curso de SQL.
- Analizamos la lógica de diferentes proyectos.



BREAK

¡5/10 MINUTOS Y VOLVEMOS!



WORKSHOP 2: EJEMPLO EN VIVO

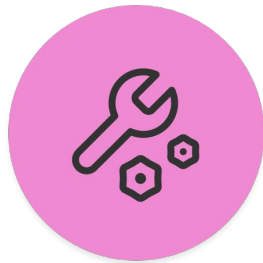
EJEMPLO EN VIVO: *Consignas a cumplir*

Ejemplo
en vivo



- Potenciar el proyecto que creamos durante el Workshop I.
- Manipular su contenido a través de **DML** (INSERT, UPDATE, DELETE).
- Integrar Subconsultas dentro de DML.
- Agregar contenido a las tablas mediante importación del mismo.
- Asegurar los datos a eliminar o actualizar mediante **UPDATE/DELETE CASCADE**.
- Integrar Funciones Almacenadas.
- Agregar Stored Procedures para facilitar el manejo de datos.
- Integrar Triggers para crear tablas de Auditoría.

EXPONER AVANCES DE PROYECTOS



COMPARTIENDO PROYECTOS

Estado de avance compañeros.

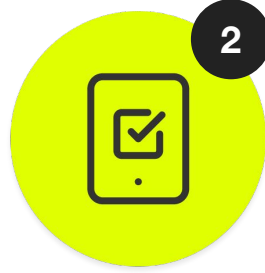
Tiempo estimado: 15 minutos



COMPARTIENDO PROYECTOS

A continuación 3 estudiantes del curso compartirán el estado de avance de sus proyectos, las dudas que le hayan surgido y recibirán feedback personalizado.

Tiempo estimado: 15 minutos



SEGUNDA ENTREGA DEL PROYECTO FINAL

Deberás entregar los avances correspondiente a la segunda entrega de tu proyecto final, incluyendo lo presentado y ajustado en la primera entrega.

PRIMERA ENTREGA DEL PROYECTO FINAL

Formato: Documento PDF con el nombre "Entrega2 + Apellido".

Sugerencia: activar la posibilidad de realizar comentarios en el archivo que subis como Entrega. Debe incluir el contenido del documento PDF de la primera entrega parcial, además de...

Proyecto
Final



2

- **Listado de Vistas** más una descripción detallada, su objetivo, y qué tablas las componen.
- **Listado de Funciones** que incluyan una descripción detallada, el objetivo para la cual fueron creadas y qué datos o tablas manipulan y/o son implementadas.
- **Listado de Stored Procedures** con una descripción detallada, qué objetivo o beneficio aportan al proyecto, y las tablas que lo componen y/o tablas con las que interactúa.

Un archivo **.sql** que contenga:

- Script de inserción de datos en las bases.
- Si se insertan datos mediante importación, agregar el paso a paso de éste en el DOC PDF más los archivos con el contenido a importar, en el formato que corresponda.
- Script de creación de Vistas, Funciones, Stored Procedures y Triggers.

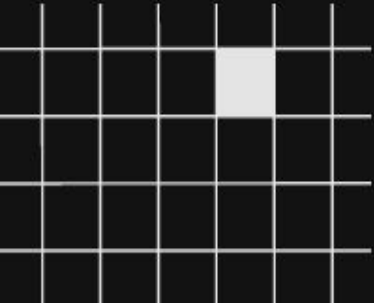
¿PREGUNTAS?





¡MUCHAS GRACIAS!

Resumen de lo visto en clase hoy:

- Repaso de Sublenguaje DML
 - DML y Subconsultas
 - Funciones almacenadas, Stored Procedures y Triggers
 - Importación y Exportación de datos
 - Actualizar y Eliminar datos en Cascada
 - Workshop II
- 



OPINA Y VALORA ESTA CLASE

#DEMOCRATIZANDO LA EDUCACIÓN

CODER HOUSE