



Clase 14. SQL

# ***ACTUALIZACIÓN Y ELIMINACIÓN DE DATOS***

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





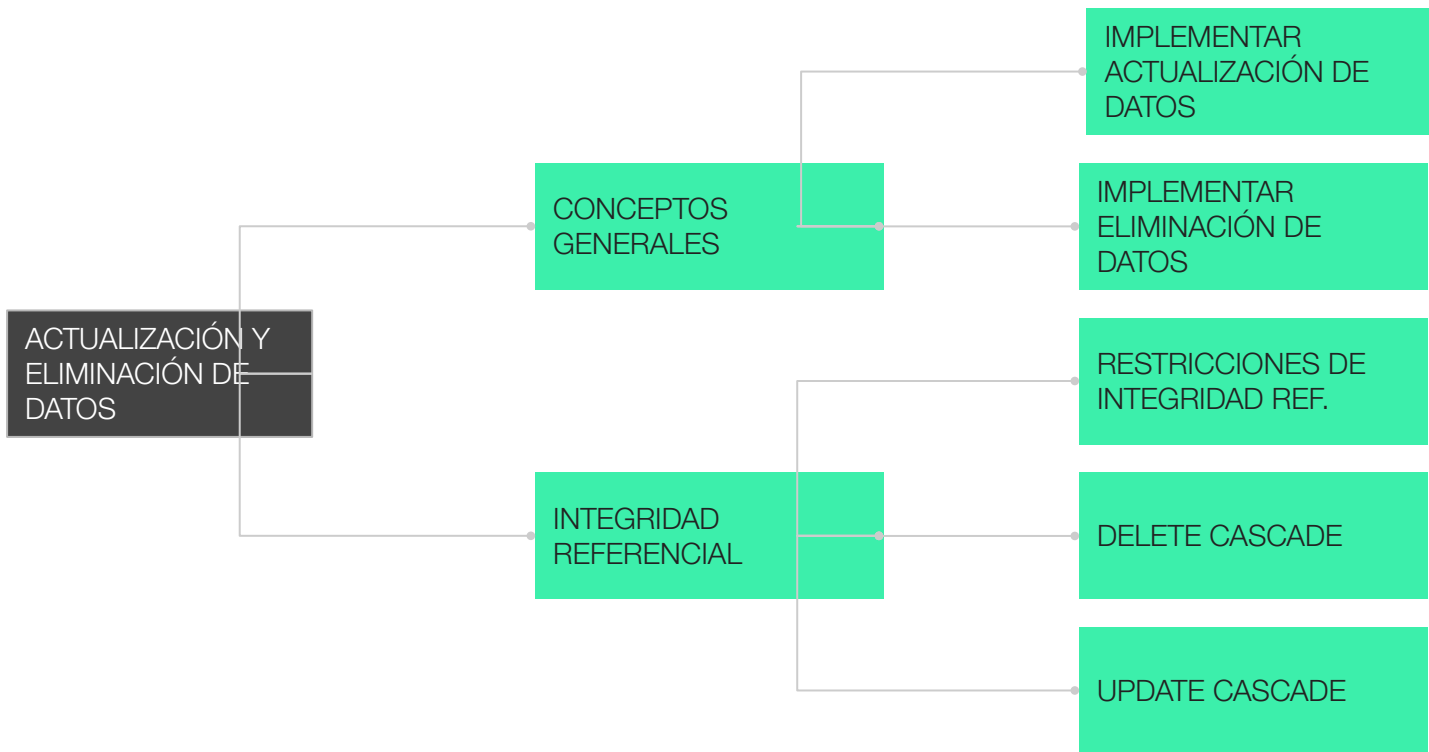
## ***OBJETIVOS DE LA CLASE***

- Reconocer la integridad referencial
- Identificar las restricciones de integridad referencial
- Implementar los procesos de delete cascade y update cascade

# ***MAPA DE CONCEPTOS***

# MAPA DE CONCEPTOS CLASE 14

¡Para  
recordar!

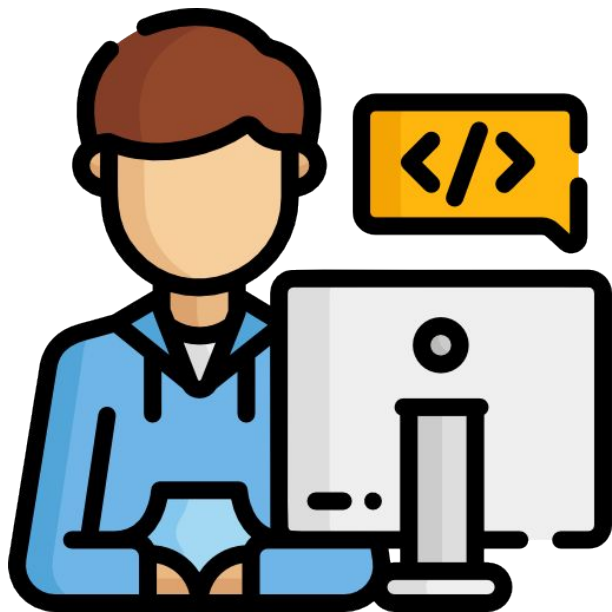


# ***INTEGRIDAD REFERENCIAL***

# ***DEFINICIÓN DE INTEGRIDAD REFERENCIAL***



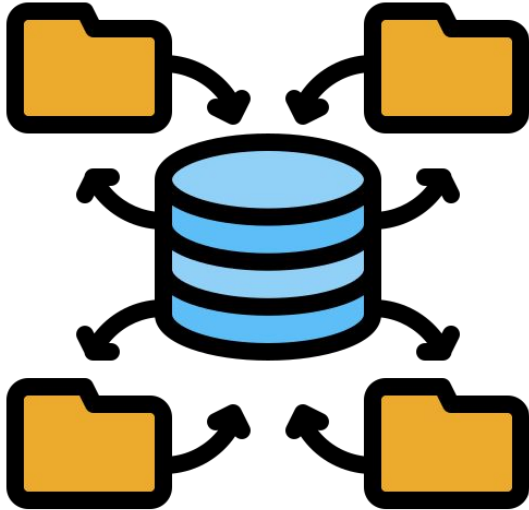
# ***INTEGRIDAD REFERENCIAL***



Continuamos aumentando la complejidad y opciones de manejo de datos que propone la base de datos relacional. Aprendamos ahora **qué es la Integridad Referencial**, y **qué importancia tiene** esta en la gestión efectiva de la información almacenada.



# ***¿QUÉ ES LA INTEGRIDAD REFERENCIAL?***



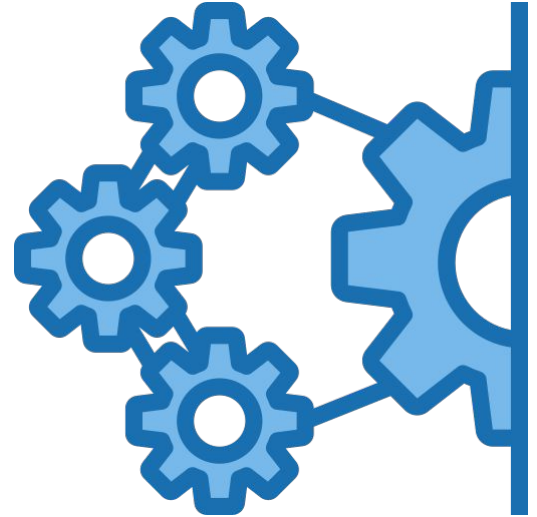
La integridad referencial garantiza que, la clave externa de una tabla de referencia, coincida siempre con una fila válida que exista en otra tabla, asegurándose así que la relación entre éstas dos tablas continúe sincronizada durante las operaciones **UPDATE** y **DELETE**.

# ***¿QUÉ ES LA INTEGRIDAD REFERENCIAL?***



Existen tres tipos de integridad referencial:

- Integridad referencial débil
- Integridad referencial parcial
- Integridad referencial completa



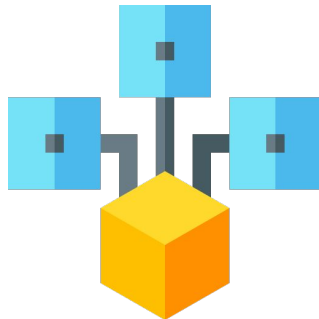
Veamos a continuación, una breve descripción de cada uno de ellos.

# ***RESTRICCIONES***



# ***INTEGRIDAD REFERENCIAL DÉBIL***

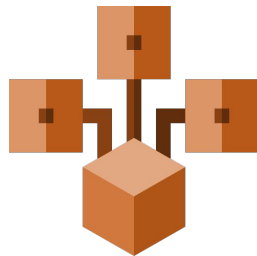
Si en una tupla de R, todos los valores de los atributos de K, poseen un valor que no es del tipo nulo, entonces debe existir una tupla en S que tome esos mismos valores en los atributos de J.





# ***INTEGRIDAD REFERENCIAL PARCIAL***

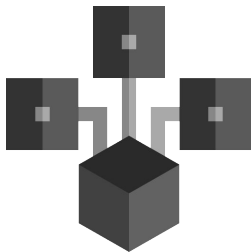
Si en una tupla de R, algún atributo de K cambia a valor nulo, entonces debe existir una tupla en S que tome en los atributos de J, los mismos valores que los atributos de K con valor nulo.





# ***INTEGRIDAD REFERENCIAL COMPLETA***

En una tupla de R, todos los atributos de K deben tener el valor nulo, o bien, todos tienen un valor que no es el nulo por lo cual debe existir una tupla en S que tome en los atributos de J, los mismos valores que toman los de K.



# ***RESTRICCIONES DE INTEGRIDAD***

# ***¿QUÉ SON LAS RESTRICCIONES DE INTEGRIDAD?***



Las restricciones de integridad se especifican en el esquema de la base de datos, debiendo ésta cumplir en cualquier extensión del mismo.

Esto hará que el Sistema de Gestión de Base de Datos vele en todo momento por la integridad de la información almacenada.



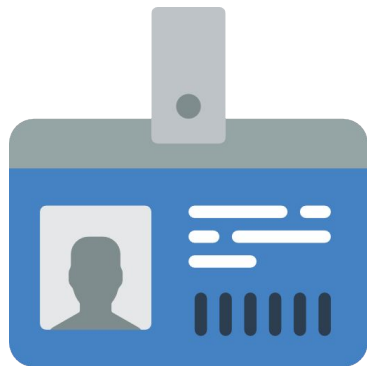
# ***TIPOS DE RESTRICCIONES***



El modelo relacional contempla cuatro tipos de restricciones:

- restricción de unicidad
- restricción de valor no nulo
- restricción de clave primaria
- restricción de integridad referencial

# ***RESTRICCIÓN DE UNICIDAD***



Pensando en una base de datos de **Empleados**, incluir en esta el atributo DNI (*documento*), permitirá identificar unívocamente a un empleado. Así se cumple la Restricción de Unicidad.

Es muy común encontrar que un DNI o Documento no es una clave unívoca.

# ***RESTRICCIÓN DE UNICIDAD***



Esto siempre se debe tener en cuenta y configurar como valor único, dado que existen posibilidades de que se repitan los Documentos de identidad. Una posibilidad fija siempre es el error humano en el tipeo de la información. Otra posibilidad es la coincidencia de documentos con el paso del tiempo.



En la Argentina, cada ciudadano extranjero que desea residir en el país, debe sacar el documento de identidad argentino en donde figuraba su nacionalidad real. Este documento de identidad otorga un número único a cada ciudadano. Pero, por ejemplo, aquellos ciudadanos que llegaron a la Argentina en época de posguerra (*2ª Guerra Mundial*), recibían su documento nacional comenzando con el número 50 millones.



Hoy, se está por alcanzar dicho número (*si es que ya no lo hizo*), otorgando documentos de identidad a los nacidos en este territorio y aún muchos ciudadanos extranjeros que llegaron hace más de 50 años al país, aún viven. Cualquier sistema informático como ser el registro nacional de las personas (*RENAPER*) o el de alguna empresa privada u organismo estatal, si no contempla la unicidad en los números de documento, podrá terminar registrando a dos personas bajo una misma identidad, causando así un error futuro en trámites personales (*aportes sociales y jubilatorios, cuentas bancarias, etcétera*).

# ***RESTRICCIÓN DE VALOR NO NULO***



Pensando nuevamente en una base de datos de **Empleados**, esta debe incluir mínimamente datos personales de un empleado donde se asiente:

- nombre y apellido
- documento de identidad
- número de legajo del empleado

# ***RESTRICCIÓN DE CLAVE PRIMARIA***



Cuando somos empleados, la organización nos otorga un número de legajo, a través del cual se asienta todo nuestro registro (*datos personales, familiares, académicos, haberes mensuales, tarjeta de acceso para presentismo, fecha de ingreso/egreso, cargo ocupado, etcétera*).

# ***RESTRICCIÓN DE INTEGRIDAD REFERENCIAL***



Y como empleados en una organización, estaremos asociados a un Departamento, Sector, o Área específica dentro de ésta.

Dicha información estará referida al atributo correspondiente de la base de datos Empleados.





# ***INTEGRIDAD REFERENCIAL***

**TIEMPO: 10 MIN**



# ***ACUERDOS***



**Presencia**



**Escucha Activa**



**Apertura al  
aprendizaje**



**Todas las voces**



# ***INTEGRIDAD REFERENCIAL***

**Consigna:** En grupos de 8-10 integrantes, se identificarán distintos ejemplos de Integridad Referencial, según los tipos descritos.

**NOTA:** usaremos los breakouts rooms. El tutor participará intermitentemente de cada conversación.



***BREAK***

**¡5/10 MINUTOS Y VOLVEMOS!**

# ***RESTRICCIONES DE INTEGRIDAD REFERENCIAL***

# ***ESTABLECER LAS CONDICIONES DE RESTRICCIÓN***

# ***ACCIONES CON RESTRICCIÓN DE INTEGRIDAD***



Cuando definimos una tabla con sus respectivos índices, claves y relaciones, tenemos la posibilidad de especificar cuatro tipo de acciones diferentes sobre las tablas relacionadas y cómo afectará la información de éstas en el momento de eliminar o actualizar el dato clave de un registro.

# ***ACCIONES CON RESTRICCIÓN DE INTEGRIDAD***



Estas acciones a definir, son:

- CASCADE
- SET NULL
- NO ACTION
- RESTRICT

Y se aplican directamente sobre las cláusulas **ON DELETE** y **ON UPDATE**.



# ***RESTRICCIONES DE INTEGRIDAD EN MYSQL***



En las bases de datos **Mysql InnoDB**, pueden especificarse las restricciones del tipo clave externa (*Foreign Key Constraints*), para que la Base de datos impida que se realicen operaciones sobre registros que provoquen inconsistencia en la información almacenada.



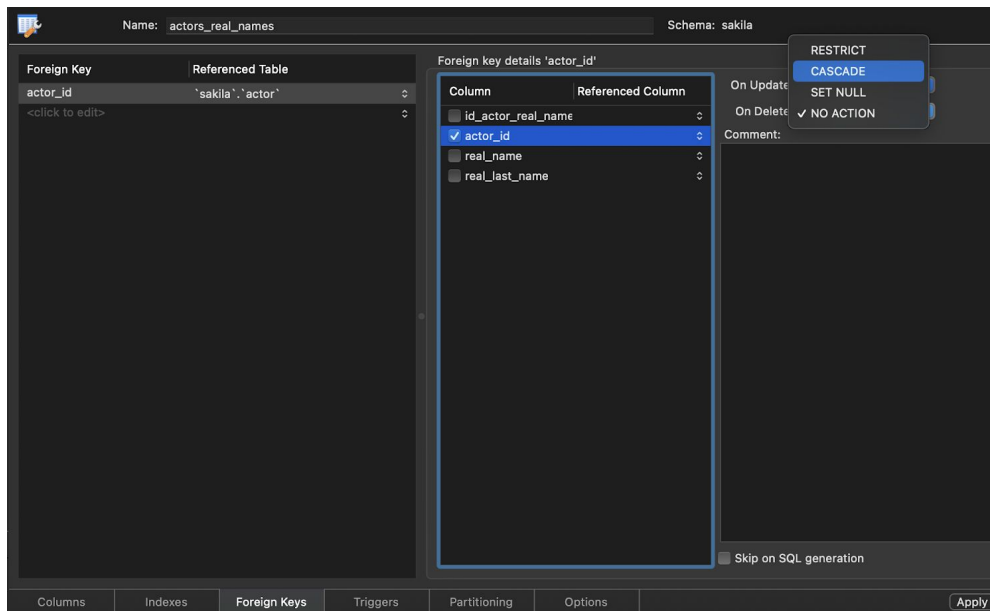
# ***RESTRICCIONES DE INTEGRIDAD EN MYSQL***



De esta forma impediremos a través de una restricción de clave externa que se cambien valores en la base de datos mediante **DELETE** o **UPDATE**, afectando de forma no controlada la integridad referencial de sus tablas.



# ***RESTRICCIONES DE INTEGRIDAD EN MYSQL***



**Mysql Workbench** tiene disponible esta opción dentro del módulo **Creación de tablas > Foreign Keys**.

# ***ELIMINACIÓN DE DATOS***

***ON DELETE CASCADE***



Ejemplo  
en vivo



# ***ON DELETE CASCADE***

Veamos un ejemplo de cómo se comportarían las acciones mencionadas ante la eliminación de datos en cascada. Pero antes creemos las tablas **PAIS** y **PERSONAS** que necesitamos:

```
CREATE TABLE PAIS (  
    pais_id INT PRIMARY KEY,  
    nombre_pais VARCHAR(50)  
);  
  
CREATE TABLE PERSONAS (  
    persona_id INT PRIMARY KEY,  
    nombre_completo VARCHAR(60),  
    pais_id INT  
);
```

***CODER HOUSE***



Ejemplo  
en vivo



# ***ON DELETE CASCADE***

Creamos la clave extranjera entre las tablas **personas** y **pais** con la restricción de integridad referencial **on delete cascade**:

```
ALTER TABLE PERSONAS ADD CONSTRAINT FK_PERSONAS_PAIS  
FOREIGN KEY (pais_id) REFERENCES PAIS(pais_id) ON DELETE CASCADE;
```



# ***ON DELETE CASCADE***

Agregamos datos a las tablas:

```
INSERT INTO PAIS (pais_id, nombre_pais)
VALUES (1, 'España'),
       (2, 'Italia'),
       (3, 'Argentina'),
       (4, 'Albania'),
       (5, 'Brasil');
```

```
INSERT INTO PERSONAS (persona_id, nombre_completo, pais_id)
VALUES (1, 'Fernando Omar', 3),
       (2, 'Julián Conte', 3),
       (3, 'Nicolás Mariano', 1),
       (4, 'Laura Grisel', 2),
       (5, 'Constantino Pascual', 4);
```





# ***TABLAS RELACIONADAS: personas / país***

Ejemplo  
en vivo



persona_id	nombre_completo	pais_id
1	Fernando Omar	3
2	Julián Conte	3
3	Nicolás Mariano	1
4	Laura Grisel	2
5	Constantino Pascual	4

pais_id	nombre_pais
1	España
2	Italia
3	Argentina
4	Albania
5	Brasil

Partimos de las siguientes tablas, donde la columna **pais\_id** de la **tabla personas** es **clave foránea** y la referencia es la columna homónima de la **tabla país**.



# ***TABLAS RELACIONADAS: personas / país***

Ejemplo  
en vivo



persona_id	nombre_completo	pais_id
1	Fernando Omar	3
2	Julián Conte	3
3	Nicolás Mariano	1
4	Laura Grisel	2
5	Constantino Pascual	4

pais_id	nombre_pais
1	España
2	Italia
3	Argentina
4	Albania
5	Brasil

Intentemos eliminar el país Argentina con la siguiente sentencia:

```
DELETE FROM PAIS WHERE pais_id = 3;
```



# ***TABLAS RELACIONADAS: personas / país***

Ejemplo  
en vivo



persona_id	nombre_completo	pais_id
3	Nicolás Mariano	1
4	Laura Grisel	2
5	Constantino Pascual	4

pais_id	nombre_pais
1	España
2	Italia
4	Albania
5	Brasil

Se eliminó el país Argentina de la tabla **PAIS** como sus referencias en la tabla **PERSONAS** debido a que se establece como acción referencial para borrado en cascada.

***ON DELETE - RESTRICT - NO ACTION***



Ejemplo  
en vivo



# ***ON DELETE RESTRICT - NO ACTION***

La acción referencia **restrict** y **no action** tienen igual resultado, aunque tienen diferentes tiempos de ejecución. La acción referencial restrict se ejecuta luego de la aplicación de la sentencia delete y no action antes de ejecutarse la sentencia delete.



Ejemplo  
en vivo



# ***ON DELETE RESTRICT - NO ACTION***

Creamos la clave extranjera entre las tablas **personas** y **pais** con la restricción de integridad referencial **on delete restrict** o **no action**:

```
ALTER TABLE PERSONAS ADD CONSTRAINT FK_PERSONAS_PAIS  
FOREIGN KEY (pais_id) REFERENCES PAIS(pais_id) ON DELETE RESTRICT;
```

○

```
ALTER TABLE PERSONAS ADD CONSTRAINT FK_PERSONAS_PAIS  
FOREIGN KEY (pais_id) REFERENCES PAIS(pais_id) ON DELETE NO ACTION;
```



# ***TABLAS RELACIONADAS: personas / país***

Ejemplo  
en vivo



persona_id	nombre_completo	pais_id
1	Fernando Omar	3
2	Julián Conte	3
3	Nicolás Mariano	1
4	Laura Grisel	2
5	Constantino Pascual	4

pais_id	nombre_pais
1	España
2	Italia
3	Argentina
4	Albania
5	Brasil

Intentemos eliminar el país Argentina con la siguiente sentencia:

```
DELETE FROM PAIS WHERE pais_id = 3;
```



# ***TABLAS RELACIONADAS: personas / país***

Ejemplo  
en vivo



Si Intentamos eliminar el país Argentina, nos brindara el siguiente error debido a que la tabla **PERSONAS** contiene referencias sobre el país argentina y la acción referencial establecida es restrict.

Output

Action Output				
#	Time	Action	Message	Duration / Fetch
1	15:33:51	DELETE FROM PAIS WHERE pais_id = 3	Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('gammers_model'. 'personas', CONSTRAINT...	0.547 sec





# ***TABLAS RELACIONADAS: personas / país***

Ejemplo  
en vivo



persona_id	nombre_completo	pais_id
1	Fernando Omar	3
2	Julián Conte	3
3	Nicolás Mariano	1
4	Laura Grisel	2
5	Constantino Pascual	4

pais_id	nombre_pais
1	España
2	Italia
3	Argentina
4	Albania
5	Brasil

Ahora, intentemos eliminar el país Brasil con la siguiente sentencia:

```
DELETE FROM PAIS WHERE pais_id = 5;
```



# ***TABLAS RELACIONADAS: personas / país***

Ejemplo  
en vivo



persona_id	nombre_completo	pais_id
1	Fernando Omar	3
2	Julián Conte	3
3	Nicolás Mariano	1
4	Laura Grisel	2
5	Constantino Pascual	4

pais_id	nombre_pais
1	España
2	Italia
3	Argentina
4	Albania

Se eliminó el país Brasil de la tabla **PAIS** debido a que en la tabla **PERSONAS** no contiene referencias sobre el país Brasil.

***ON DELETE - SET NULL***



Ejemplo  
en vivo



# ***ON DELETE SET NULL***

Creamos la clave extranjera entre las tablas **personas** y **pais** con la restricción de integridad referencial **on delete set null**:

```
ALTER TABLE PERSONAS ADD CONSTRAINT FK_PERSONAS_PAIS  
FOREIGN KEY (pais_id) REFERENCES PAIS(pais_id) ON DELETE SET NULL;
```



# TABLAS RELACIONADAS: *personas / país*

Ejemplo  
en vivo



persona_id	nombre_completo	pais_id
1	Fernando Omar	3
2	Julián Conte	3
3	Nicolás Mariano	1
4	Laura Grisel	2
5	Constantino Pascual	4

pais_id	nombre_pais
1	España
2	Italia
3	Argentina
4	Albania
5	Brasil

Intentemos eliminar el país Argentina con la siguiente sentencia:

```
DELETE FROM PAIS WHERE pais_id = 3;
```



# ***TABLAS RELACIONADAS: personas / país***

Ejemplo  
en vivo



persona_id	nombre_completo	pais_id
1	Fernando Omar	NULL
2	Julián Conte	NULL
3	Nicolás Mariano	1
4	Laura Grisel	2
5	Constantino Pascual	4

pais_id	nombre_pais
1	España
2	Italia
4	Albania
5	Brasil

Al eliminar el país Argentina, se eliminará efectivamente el registro dentro de la tabla PAIS, y además se establece **NULL** a sus referencias en la tabla **PERSONAS** debido a que se estableció un acción referencial para borrado **set null**.



En Mysql, las **claves externas y columnas generadas mediante Alias o desde campos calculados**, pueden tener diferencias respecto a cómo se comportará este motor respecto al motor SQL original. Puedes consultar esto [desde la documentación original de Mysql](#).

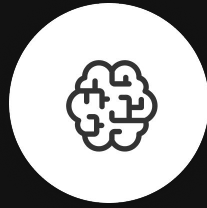
# ***IMPLEMENTAR ACTUALIZACIÓN DE DATOS***





# ***ON UPDATE***

El comportamiento en general a través de la sentencia UPDATE, es similar a lo representado en las diferentes opciones de la sentencia ON DELETE.

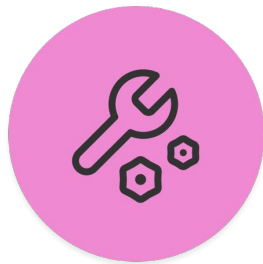


***¡PARA PENSAR!***

*¿Cómo debemos resolver la nacionalidad de personas cuyo país cambia de nombre?*

*¿Y cuando un país se divide, como sucedió con Serbia y Montenegro en 2006?*





# ***PRACTICA SQL: ON DELETE + ON UPDATE***

Practicarás los procesos de **on delete** y **on update**

Tiempo estimado: 20 minutos



Desarrolla un ejemplo similar, partiendo de dos tablas:

## Países y Localidades

Luego:

- Crea registros para, al menos, 4 países.
- Relaciona al menos 2 localidades por cada país.
- Crea las tablas utilizando la **CONSTRAINT** (**SET NULL** y **CASCADE**).
- Ejecuta una consulta de eliminación por cada **CONSTRAINT**.



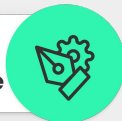
# ***PRESENTACIÓN DE SCRIPT DE CREACIÓN DE VISTAS***

Creación de 5 vistas con base en los datos de la base del proyecto final.

# PRESENTACIÓN DE SCRIPT DE CREACIÓN DE VISTAS

**Formato:** El archivo a presentar debe ser del tipo .sql nombrado como "ScriptVis+Apellido".

Desafío  
entregable



**>> Consigna:** Presentar en formato .sql el script de creación de 5 vistas con base en los datos de la base de datos del proyecto final.

**>>Aspectos a incluir en el entregable:**

En base a las tablas elegidas para tu proyecto, deberás idear 5 o más vistas de una o más tablas y generar el SCRIPT de creación de las mismas.

**>>Ejemplo:** si tu proyecto tiene una tabla productos, puedes crear una vista relacionándola con la tabla categorías, mostrar el costo de cada producto, representando distintos valores de IVA (TAX), etcétera.

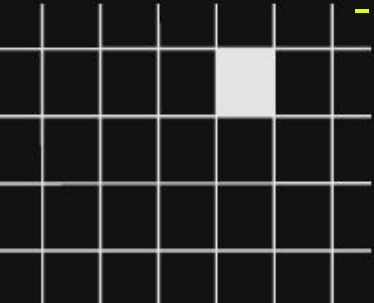
***¿PREGUNTAS?***





# ***¡MUCHAS GRACIAS!***

Resumen de lo visto en clase hoy:

- Integridad referencial.
  - Restricciones de I.R.ON
  - Delete Cascade.
  - ON Update Cascade.
- 





***OPINA Y VALORA ESTA CLASE***

***#DEMOCRATIZANDO LA EDUCACIÓN***

***CODER HOUSE***