**Repaso:** Una *clave extranjera* (**FOREIGN KEY** en SQL) de una tabla A (**referenciante**) es un conjunto de columnas cuyos valores debe coincidir al menos con uno de los valores de otro conjunto de columnas, que son *clave primaria* (**PRIMARY KEY** en SQL) de otra tabla B (**referenciada**) o bien ser nulos.

A lo anterior se lo denomina Restricción de Integridad Referencial (RIR) y en SQL se especifica en las sentencias CREATE TABLE o ALTER TABLE

**CONSTRAINT FK\_nom FOREIGN KEY (lista\_columnasFK)**

**REFERENCES nombreTablaRef [(lista\_columnasRef)]**

**[ MATCH {FULL | PARTIAL | SIMPLE}]**

**[ON UPDATE AccionRef] AccionRef = NO ACTION | CASCADE |**

**[ON DELETE AccionRef] ] } ….. ); SET NULL | SET DEFAULT | RESTRICT**

### **Si tengo algo que lo referencie no puedo borrar con delete**

tengo q seguir todo el camino

### **Ejercicio 1**

Dado el esquema de la figura: (el script lo podes descargar de [aquí](https://drive.google.com/file/d/1FhL5RggPDG1Rr8VJ-sTnfjtSZPAKZeeN/view?usp=sharing) )

1. Cómo debería implementar las Restricciones de Integridad Referencial (**RIR**) si se desea que cada vez que se elimine un registro de la tabla PALABRA , también se eliminen los artículos que la referencian en la tabla CONTIENE.

-- Table: P5P1E1\_CONTIENE

CREATE TABLE P5P1E1\_CONTIENE (

id\_articulo int NOT NULL,

idioma char(2) NOT NULL,

cod\_palabra int NOT NULL,

CONSTRAINT P5P1E1\_CONTIENE\_pk PRIMARY KEY (id\_articulo,idioma,cod\_palabra)

);

-- Table: P5P1E1\_PALABRA

CREATE TABLE P5P1E1\_PALABRA (

idioma char(2) NOT NULL,

cod\_palabra int NOT NULL,

descripcion varchar(25) NOT NULL,

CONSTRAINT P5P1E1\_PALABRA\_pk PRIMARY KEY (idioma,cod\_palabra)

);

-- foreign keys

-- Reference: FK\_P5P1E1\_CONTIENE\_ARTICULO (table: P5P1E1\_CONTIENE)

ALTER TABLE P5P1E1\_CONTIENE ADD CONSTRAINT FK\_P5P1E1\_CONTIENE\_ARTICULO

FOREIGN KEY (id\_articulo)

REFERENCES P5P1E1\_ARTICULO (id\_articulo)

ON DELETE CASCADE

NOT DEFERRABLE

INITIALLY IMMEDIATE

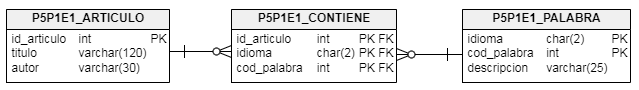
;

1. Verifique qué sucede con las palabras contenidas en cada artículo, al eliminar una palabra, si definen la Acción Referencial para las bajas (ON DELETE) de la RIR correspondiente como:

ii) Restrict: Es el comportamiento por defecto, que impide realizar modificaciones que atentan contra la integridad referencial. No permite borrar un registro cuya clave primaria está siendo referenciada por un registro en la Tabla contiene.

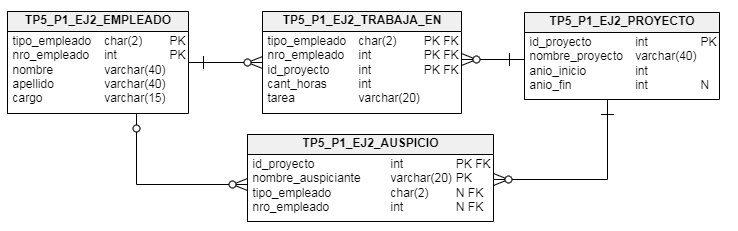
iii) Es posible para éste ejemplo colocar SET NULL o SET DEFAULT para ON DELETE y ON UPDATE?

SET NULL no se puede utilizar porque no admite nulos.

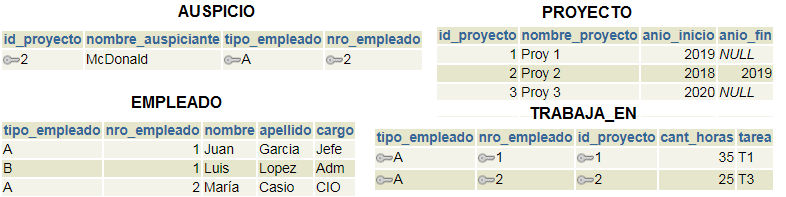


### **Ejercicio 2**

Considere el siguiente esquema de una base de datos de un Centro de Desarrollo, cuyo script de creación lo podes descargar de [aquí](https://drive.google.com/file/d/1KmYAApC8Tw2xw1z8qVGDyHNTiU8yH6IM/view?usp=sharing)

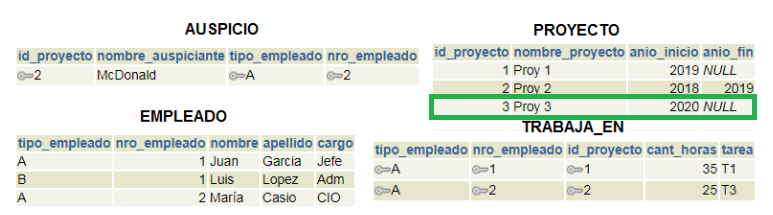


Teniendo en cuenta las cláusulas ON UPDATE y ON DELETE las FOREIGN KEY y siguientes los siguientes registros en las respectivas tablas cuyos inserts los podes descargar de [aquí](https://drive.google.com/file/d/1ejTYHRJTzQw5wuISLrK1iJj-rb_iIF56/view?usp=sharing)



* 1. Indique el resultado de las siguientes operaciones, teniendo en cuenta las acciones referenciales e instancias dadas. En caso de que la operación no se pueda realizar, indicar qué regla/s entra/n en conflicto y cuál es la causa. En caso de que sea aceptada, comente el resultado que produciría (NOTA: en cada caso considere el efecto sobre la instancia original de la BD, los resultados no son acumulativos).

b.1) delete from tp5\_p1\_ej2\_proyecto where id\_proyecto = 3;



-- Reference: FK\_TP5\_P1\_EJ2\_AUSPICIO\_PROYECTO (table: TP5\_P1\_EJ2\_AUSPICIO)

ALTER TABLE TP5\_P1\_EJ2\_AUSPICIO ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_AUSPICIO\_PROYECTO

FOREIGN KEY (id\_proyecto)

REFERENCES TP5\_P1\_EJ2\_PROYECTO (id\_proyecto)

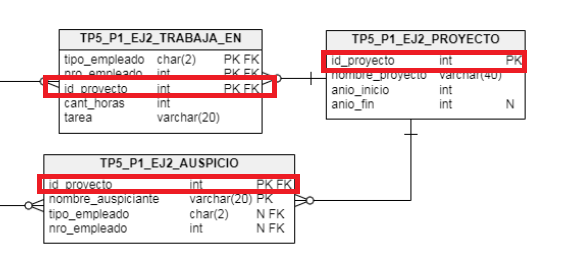
ON DELETE RESTRICT

ON UPDATE RESTRICT

;

No tengo referencias en las otras tablas de id\_3 entonces no aplico la sentencia restrictiva. (Ok)

b.2) update tp5\_p1\_ej2\_proyecto set id\_proyecto = 7 where id\_proyecto = 3;



-- Reference: FK\_TP5\_P1\_EJ2\_AUSPICIO\_PROYECTO (table: TP5\_P1\_EJ2\_AUSPICIO)

ALTER TABLE TP5\_P1\_EJ2\_AUSPICIO ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_AUSPICIO\_PROYECTO

FOREIGN KEY (id\_proyecto)

REFERENCES TP5\_P1\_EJ2\_PROYECTO (id\_proyecto)

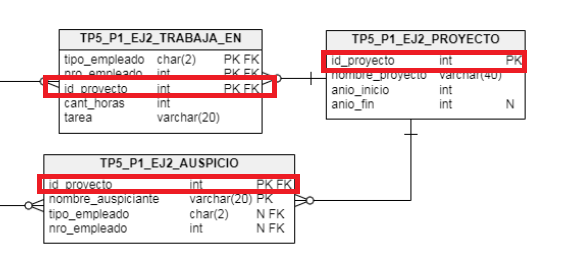
ON DELETE RESTRICT

ON UPDATE RESTRICT

;

Se puede porque id\_3 no está referenciado en las otras tablas (no existe) entonces no entra en la sentencia restrictiva.

b.3) delete from tp5\_p1\_ej2\_proyecto where id\_proyecto = 1;

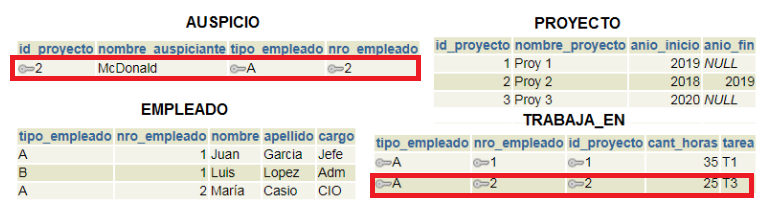


No se puede eliminar la fila. La restricción delete restrict no me deja borrar y las tablas están referenciadas.

[2023-05-08 19:54:53] [23503] ERROR: update or delete on table "tp5\_p1\_ej2\_proyecto" violates foreign key constraint "fk\_tp5\_p1\_ej2\_trabaja\_en\_proyecto" on table "tp5\_p1\_ej2\_trabaja\_en"

[2023-05-08 19:54:53] Detail: Key (id\_proyecto)=(1) is still referenced from table "tp5\_p1\_ej2\_trabaja\_en".

b.4) delete from tp5\_p1\_ej2\_empleado where tipo\_empleado = ‘A’ and nro\_empleado = 2;



-- Table: TP5\_P1\_EJ2\_EMPLEADO

CREATE TABLE TP5\_P1\_EJ2\_EMPLEADO (

tipo\_empleado char(2) NOT NULL,

nro\_empleado int NOT NULL,

nombre varchar(40) NOT NULL,

apellido varchar(40) NOT NULL,

cargo varchar(15) NOT NULL,

CONSTRAINT TP5\_P1\_EJ2\_EMPLEADO\_pk PRIMARY KEY (tipo\_empleado,nro\_empleado)

);

-- Reference: FK\_TP5\_P1\_EJ2\_AUSPICIO\_EMPLEADO (table: TP5\_P1\_EJ2\_AUSPICIO)

ALTER TABLE TP5\_P1\_EJ2\_AUSPICIO ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_AUSPICIO\_EMPLEADO

FOREIGN KEY (tipo\_empleado, nro\_empleado)

REFERENCES TP5\_P1\_EJ2\_EMPLEADO (tipo\_empleado, nro\_empleado)

MATCH FULL

ON DELETE SET NULL

ON UPDATE RESTRICT

;

Sí, se puede

no sé por qué??

MATCH FULL no permitirá que una columna de una clave foránea de varias columnas sea nula a menos que todas las columnas de la clave foránea sean nulas;

b.5) update tp5\_p1\_ej2\_trabaja\_en set id\_proyecto = 3 where id\_proyecto =1;

-- Table: TP5\_P1\_EJ2\_TRABAJA\_EN

CREATE TABLE TP5\_P1\_EJ2\_TRABAJA\_EN (

tipo\_empleado char(2) NOT NULL,

nro\_empleado int NOT NULL,

id\_proyecto int NOT NULL,

cant\_horas int NOT NULL,

tarea varchar(20) NOT NULL,

CONSTRAINT TP5\_P1\_EJ2\_TRABAJA\_EN\_pk PRIMARY KEY (tipo\_empleado,nro\_empleado,id\_proyecto)

);

-- Reference: FK\_TP5\_P1\_EJ2\_TRABAJA\_EN\_PROYECTO (table: TP5\_P1\_EJ2\_TRABAJA\_EN)

ALTER TABLE TP5\_P1\_EJ2\_TRABAJA\_EN ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_TRABAJA\_EN\_PROYECTO

FOREIGN KEY (id\_proyecto)

REFERENCES TP5\_P1\_EJ2\_PROYECTO (id\_proyecto)

ON DELETE RESTRICT

ON UPDATE CASCADE

;

-- Reference: FK\_TP5\_P1\_EJ2\_AUSPICIO\_PROYECTO (table: TP5\_P1\_EJ2\_AUSPICIO)

ALTER TABLE TP5\_P1\_EJ2\_AUSPICIO ADD CONSTRAINT FK\_TP5\_P1\_EJ2\_AUSPICIO\_PROYECTO

FOREIGN KEY (id\_proyecto)

REFERENCES TP5\_P1\_EJ2\_PROYECTO (id\_proyecto)

ON DELETE RESTRICT

ON UPDATE RESTRICT

;

Se puede modificar id\_3 porque no está referenciado (no existe en las otras tablas).

b.6) update tp5\_p1\_ej2\_proyecto set id\_proyecto = 5 where id\_proyecto = 2;

UPDATE nombre\_tabla SET columna1 = 'nuevo\_valor' WHERE columna1 = 'valor1';

[23503] ERROR: update or delete on table "tp5\_p1\_ej2\_proyecto" violates foreign key constraint "fk\_tp5\_p1\_ej2\_auspicio\_proyecto" on table "tp5\_p1\_ej2\_auspicio" Detail: Key (id\_proyecto)=(2) is still referenced from table "tp5\_p1\_ej2\_auspicio".

* 1. Indique el resultado de la siguiente operaciones justificando su elección:

update auspicio set id\_proyecto= 66, nro\_empleado = 10

where id\_proyecto = 22

and tipo\_empleado = 'A'

and nro\_empleado = 5;

(suponga que existe la tupla asociada)

-- Table: TP5\_P1\_EJ2\_AUSPICIO

CREATE TABLE TP5\_P1\_EJ2\_AUSPICIO (

id\_proyecto int NOT NULL,

nombre\_auspiciante varchar(20) NOT NULL,

tipo\_empleado char(2) NULL,

nro\_empleado int NULL,

CONSTRAINT TP5\_P1\_EJ2\_AUSPICIO\_pk PRIMARY KEY (id\_proyecto,nombre\_auspiciante)

);

Se puede (OK)

i. realiza la modificación si existe el proyecto 22 y el empleado TipoE = 'A' ,NroE = 5 (X)

ii. realiza la modificación si existe el proyecto 22 sin importar si existe el empleado TipoE = 'A' ,NroE = 5

iii.se modifican los valores, dando de alta el proyecto 66 en caso de que no exista (si no se violan restricciones de nulidad), sin importar si existe el empleado

iv. se modifican los valores, y se da de alta el proyecto 66 y el empleado correspondiente (si no se violan restricciones de nulidad)

v. no permite en ningún caso la actualización debido a la modalidad de la restricción entre la tabla empleado y auspicio.

vi. ninguna de las anteriores, cuál?

--Primero agrego el dato

insert into tp5\_p1\_ej2\_empleado (tipo\_empleado, nro\_empleado, nombre, apellido, cargo) VALUES ('B', 10, 'Feli', 'Ague', 'jefe');

insert into tp5\_p1\_ej2\_proyecto (id\_proyecto, nombre\_proyecto, anio\_inicio, anio\_fin) values (66, 'NN', 2022, 2023);

insert into tp5\_p1\_ej2\_auspicio (id\_proyecto, nombre\_auspiciante, tipo\_empleado, nro\_empleado) VALUES (66, 'Mc Donald', 'B', 10);

--ahora hago la consulta

update tp5\_p1\_ej2\_auspicio set id\_proyecto= 66, nro\_empleado = 10

where id\_proyecto = 22

and tipo\_empleado = 'A'

and nro\_empleado = 5;

--luego reviso lo que modifique

select \* from tp5\_p1\_ej2\_auspicio where id\_proyecto=66;

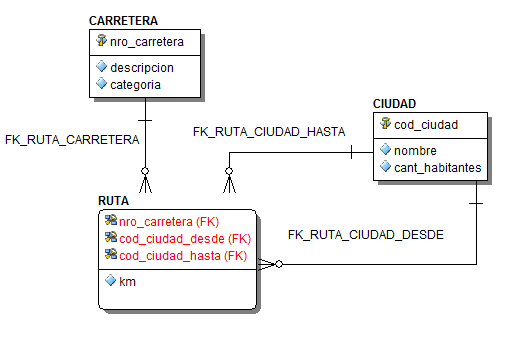
* 1. Indique cuáles de las siguientes operaciones serán aceptadas/rechazadas, según se considere para las relaciones AUSPICIO-EMPLEADO y AUSPICIO-PROYECTO match: i) simple, ii) parcial, o iii) full:

| La integridad referencial se satisface si para cada registro en la tabla  referenciante se verifica lo siguiente:  1- Ninguno de los valores de las columnas de la FK es NULL y existe  un registro en la tabla referenciada cuyos valores de clave coinciden con  los de tales columnas,  2- o  Al menos un valor en una de las columnas de la FK es NULL y puede o NO  el resto de los valores de las columnas hacer referencia a la PK (MATCH  SIMPLE)  Los valores de las columnas no nulos de la FK se corresponden con los  correspondientes valores de clave en al menos en un registro de la tabla  referenciada (MATCH PARTIAL)  Todas las columnas de la FK son NULL (MATCH FULL) o hacen referencia  a la PK completa |
| --- |

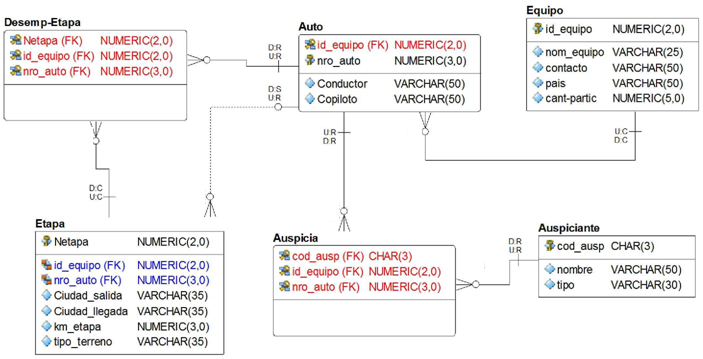
* + 1. insert into Auspicio values (1, Dell , B, null);
    2. insert into Auspicio values (2, Oracle, null, null);
    3. insert into Auspicio values (3, Google, A, 3);
    4. insert into Auspicio values (1, HP, null, 3);

### **Ejercicio 3.**

Sea el siguiente DERExt :

1. Se podrá declarar como acción referencial de la (RIR) *FK\_Ruta\_ciudad\_desde* DELETE CASCADE y para la RIR *FK\_Ruta\_ciudad\_hasta*  DELETE RESTRICT ?
2. Es posible colocar DELETE SET NULL o UPDATE SET NULL como acción referencial de la RIR FK\_Ruta\_Carretera ?

### **Ejercicio 4.**

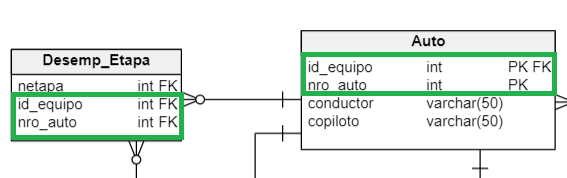


1. Es posible definir las siguientes RIRs tal como se declaran en cada punto? Indique V o F según corresponda y justifique.
2. ALTER TABLE Desemp-Etapa

ADD CONSTRAINT FK\_DesempEtapa\_Auto FOREIGN KEY (id\_equipo)

REFERENCES Auto (id\_equipo);

(F) Falta referenciar la PK nro\_auto.



-- Reference: Desemp\_Etapa\_Auto (table: Desemp\_Etapa)

ALTER TABLE Desemp\_Etapa ADD CONSTRAINT Desemp\_Etapa\_Auto

FOREIGN KEY (nro\_auto, id\_equipo)

REFERENCES Auto (nro\_auto, id\_equipo)

NOT DEFERRABLE

INITIALLY IMMEDIATE

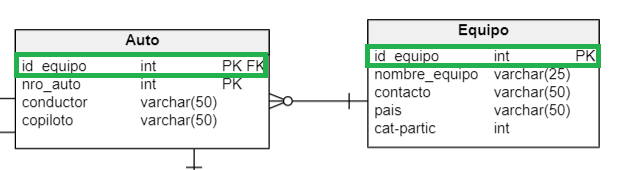
;

1. ALTER TABLE Equipo

ADD CONSTRAINT FK\_Equipo\_Auto FOREIGN KEY (id\_equipo, contacto)

REFERENCES Auto (id\_equipo, conductor);

(F) La única PK que se une a equipo es id\_equipo.



ALTER TABLE Auto ADD CONSTRAINT Auto\_Equipo

FOREIGN KEY (id\_equipo)

REFERENCES Equipo (id\_equipo)

NOT DEFERRABLE

INITIALLY IMMEDIATE

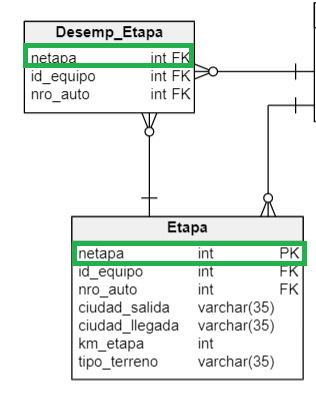
;

1. ALTER TABLE Desemp-Etapa

ADD CONSTRAINT FK\_DesempEtapa\_Etapa FOREIGN KEY (Netapa, id\_equipo)

REFERENCES Etapa (etapa, id\_equipo);

(F) La PK de ETAPA(netapa) referencia a la FK de DESEMP\_ETAPA (netapa)



ALTER TABLE Desemp\_Etapa ADD CONSTRAINT Desemp\_Etapa\_Etapa

FOREIGN KEY (netapa)

REFERENCES Etapa (netapa)

NOT DEFERRABLE

INITIALLY IMMEDIATE

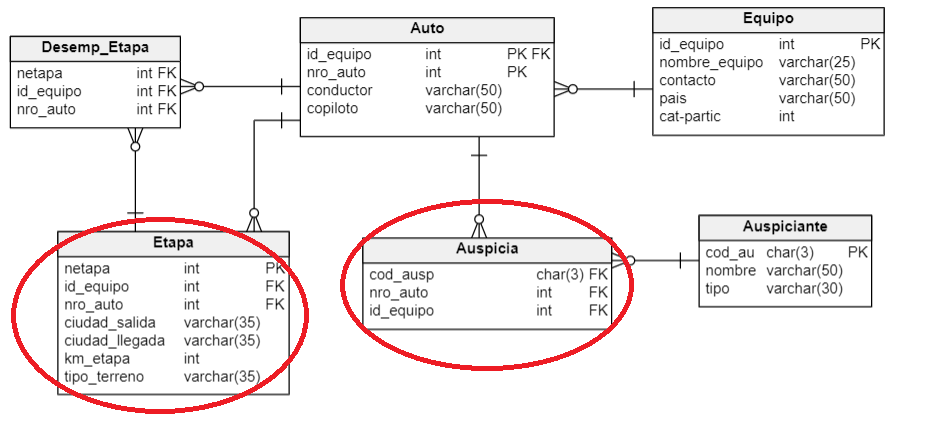
;

1. ALTER TABLE Auspicia

ADD CONSTRAINT FK\_Auspicia\_Etapa FOREIGN KEY (nro\_auto)

REFERENCES Etapa (nro\_auto);

(F) La tabla Auspicia con Etapa no está relacionada.

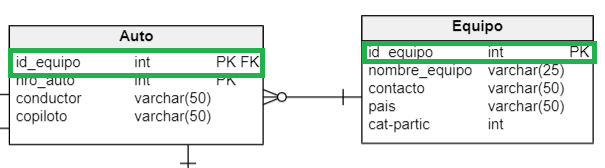


1. ALTER TABLE Auto

ADD CONSTRAINT FK\_Auto\_Equipo FOREIGN KEY (id\_equipo)

REFERENCES Equipo (id\_equipo);

(V) La PK id\_equipo de la tabla EQUIPO es la única que referencia a la FK (id\_equipo) de la tabla AUTO



-- Reference: Auto\_Equipo (table: Auto)

ALTER TABLE Auto ADD CONSTRAINT Auto\_Equipo

FOREIGN KEY (id\_equipo)

REFERENCES Equipo (id\_equipo)

NOT DEFERRABLE

INITIALLY IMMEDIATE

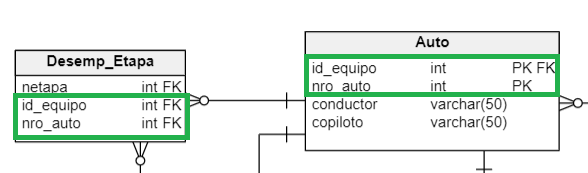
;

1. ALTER TABLE Desemp-Etapa

ADD CONSTRAINT FK\_DesempEtapa\_Auto FOREIGN KEY (id\_equipo, nro\_auto)

REFERENCES Auto (id\_equipo, nro\_auto);

(V) Las PK de auto (id\_equipo y nro\_auto) referencian a la tabla DESEMP\_AUTO.



-- Reference: Desemp\_Etapa\_Auto (table: Desemp\_Etapa)

ALTER TABLE Desemp\_Etapa ADD CONSTRAINT Desemp\_Etapa\_Auto

FOREIGN KEY (nro\_auto, id\_equipo)

REFERENCES Auto (nro\_auto, id\_equipo)

NOT DEFERRABLE

INITIALLY IMMEDIATE

;