

---

# Restricciones de Clave Foránea en PostgreSQL

## 1. A Nivel de Columna

Cuando defines la restricción directamente en la definición de la columna, puedes usar la cláusula **CONSTRAINT** para asignarle un nombre.

### Código Completo

-- Crear la tabla referenciada

```
CREATE TABLE departamentos (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(50) NOT NULL  
);
```

-- Crear la tabla que contiene la clave foránea

```
CREATE TABLE empleados (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    id_departamento INT CONSTRAINT fk_empleados_departamentos  
REFERENCES departamentos(id)  
);
```

- **Tabla Referenciada (departamentos):**
    - Tiene una clave primaria **id**, que será referenciada por otras tablas.
  - **Tabla Contenedora de Clave Foránea (empleados):**
    - La columna **id\_departamento** contiene una restricción **FOREIGN KEY** que referencia el campo **id** de **departamentos**.
    - La restricción se llama **fk\_empleados\_departamentos**.
-

## 2. A Nivel de Tabla

Si prefieres definir la restricción fuera de la definición de la columna, puedes hacerlo a nivel de tabla. Esto es útil cuando se requiere un código más descriptivo o al trabajar con claves compuestas.

### Código Completo

-- Crear la tabla referenciada

```
CREATE TABLE cursos (  
    id SERIAL PRIMARY KEY,  
    titulo VARCHAR(100) NOT NULL  
);
```

-- Crear la tabla que contiene la clave foránea

```
CREATE TABLE modulos (  
    id SERIAL PRIMARY KEY,  
    titulo VARCHAR(100) NOT NULL,  
    id_curso INT,  
    CONSTRAINT fk_modulos_cursos FOREIGN KEY (id_curso) REFERENCES  
cursos(id)  
);
```

- **Tabla Referenciada (**cursos**):**
  - Contiene el campo **id** como clave primaria.
- **Tabla Contenedora de Clave Foránea (**modulos**):**
  - La columna **id\_curso** referencia la clave primaria **id** de la tabla **cursos**.
  - La restricción se llama **fk\_modulos\_cursos**.

---

## 3. Clave Foránea Compuesta

Las claves foráneas compuestas se utilizan cuando la relación depende de más de una columna. Esto solo se puede definir a nivel de tabla.

### Código Completo

-- Crear la tabla referenciada

```
CREATE TABLE cursos_estudiantes (  
    id_estudiante INT,  
    id_curso INT,  
    PRIMARY KEY (id_estudiante, id_curso)  
);
```

-- Crear la tabla que contiene la clave foránea compuesta

```
CREATE TABLE evaluaciones (  
    id SERIAL PRIMARY KEY,  
    id_estudiante INT,  
    id_curso INT,  
    nota DECIMAL(5,2) CHECK (nota >= 0 AND nota <= 10),  
    CONSTRAINT fk_evaluaciones_cursos_estudiantes FOREIGN KEY  
(id_estudiante, id_curso)  
    REFERENCES cursos_estudiantes(id_estudiante, id_curso)  
);
```

- **Tabla Referenciada (**cursos\_estudiantes**):**
  - Tiene una clave primaria compuesta por las columnas **id\_estudiante** e **id\_curso**.
- **Tabla Contenedora de Clave Foránea (**evaluaciones**):**
  - La combinación de **id\_estudiante** e **id\_curso** debe coincidir con un registro existente en **cursos\_estudiantes**.
  - La restricción se llama **fk\_evaluaciones\_cursos\_estudiantes**.

---

## 4. Configuración de Acciones: **ON DELETE** y **ON UPDATE**

Las restricciones de clave foránea pueden incluir configuraciones que definan qué ocurre al eliminar o actualizar registros en la tabla referenciada.

### Código Completo con Acciones

-- Crear la tabla referenciada

```
CREATE TABLE departamentos (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(50) NOT NULL  
);
```

-- Crear la tabla que contiene la clave foránea con acciones

```
CREATE TABLE empleados (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    id_departamento INT,  
    CONSTRAINT fk_empleados_departamentos FOREIGN KEY  
    (id_departamento)  
    REFERENCES departamentos(id)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE  
);
```

- **Acciones Definidas:**

- **ON DELETE SET NULL:** Si un departamento es eliminado, el campo `id_departamento` en `empleados` se establece como `NULL`.

- **ON UPDATE CASCADE:** Si el **id** de un departamento cambia, el valor correspondiente en **empleados** también se actualiza.
- 

## 5. Modificar Tablas Existentes

Si las tablas ya están creadas y necesitas añadir una clave foránea, puedes usar **ALTER TABLE**.

### Ejemplo

-- Crear la tabla referenciada

```
CREATE TABLE categorias (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(50) NOT NULL  
);
```

-- Crear la tabla sin la clave foránea

```
CREATE TABLE productos (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    id_categoria INT  
);
```

-- Añadir la clave foránea posteriormente

```
ALTER TABLE productos  
  
ADD CONSTRAINT fk_productos_categorias FOREIGN KEY (id_categoria)  
REFERENCES categorias(id);
```

---

# Consulta de Restricciones

Puedes consultar las claves foráneas definidas en las tablas usando las siguientes vistas del catálogo del sistema:

## Ver Todas las Claves Foráneas en la Base de Datos

```
SELECT conname AS nombre_restriccion,
       conrelid::regclass AS tabla,
       confrelid::regclass AS tabla_referenciada
FROM pg_constraint
WHERE contype = 'f';
```

## Ver Claves Foráneas de una Tabla Específica

```
SELECT conname AS nombre_restriccion,
       confrelid::regclass AS tabla_referenciada
FROM pg_constraint
WHERE contype = 'f' AND conrelid = 'productos'::regclass;
```

---

## Resumen

Nivel	Ventaja	Uso Común
A nivel de columna	Más compacto y directo.	Para claves foráneas simples en una sola columna.
A nivel de tabla	Más descriptivo y necesario para claves foráneas compuestas o con configuraciones avanzadas.	Para claves compuestas o con múltiples opciones adicionales.

