

## TALLER II: RESTRICCIONES DE INTEGRIDAD EN SQL

## Objetivos

- Definir restricciones de integridad de entidad a través de la *clave primaria*.
- Definir restricciones de integridad referencial a través de la *clave foránea*.
- Reconocer violaciones a las restricciones de integridad.
- Comprobar la efectividad de la definición de restricciones.
- Familiarizarse con las siguientes cláusulas del lenguaje SQL:

PRIMARY KEY		CONSTRAINT
FOREIGN KEY		REFERENCES
ON DELETE		ON UPDATE

- (*Establecimiento de una conexión*) Ingrese al administrador *pgAdmin* y conéctese a la base **Mundial** del servidor de *PostgreSQL* local a su computadora.

## Si no tienen la base...

Utilizaremos la base **Mundial** que creamos durante el taller anterior, y sus tablas **teams** y **matches** con sus datos ya insertados. Si no tiene esta base, utilice el *Script* que se provee en el Campus junto a este enunciado para generarla:

- Cree una base nueva de nombre *Mundial*.
- Abra el *Script* con el *Query Tool* del *pgAdmin4*.
- Ejecute el script. El mismo creará las tablas e insertará los datos en ellas. Recuerde especificar correctamente las rutas a los archivos csv.
  - `teams(team, players_used, avg_age, possession, games, goals, assists, cards_yellow, cards_red)`
  - `matches(team1, team2, goals.team1, goals.team2, stage)`

- (*Restricciones de clave primaria*) De acuerdo con el modelo relacional propuesto, qué atributos conforman la clave primaria de cada tabla. Abra el *Script* que se provee junto a este enunciado en el editor SQL (*Query Tool*) del *pgAdmin*, y modifique cada `CREATE TABLE` definiendo una `CONSTRAINT` de `PRIMARY KEY`. Asigne un nombre a cada *constraint*: `PK_teams` y `PK_matches`. Ejecute el *script*.
- (*Verificación I*) Intente provocar una violación a la restricción de unicidad de la tabla **teams** a través de un `INSERT`.
- (*Restricciones de clave foránea*) Analice en cuál de las dos tablas existe una clave foránea, y qué atributos la conforman. Modifique el `CREATE TABLE` respectivo en el *script*, definiendo una `CONSTRAINT` de `FOREIGN KEY`. Asigne un nombre a la *constraint* y ejecute el script.
- (*Verificación II: inserción de una referencia a una fila no existente*) Intente provocar una violación a la restricción de integridad referencial definida, a través de un `INSERT` en la tabla que considere apropiada.

6. (*Verificación III: eliminación de una tupla que es referenciada*) Intente provocar una violación a la restricción de integridad referencial definida, a través de un `DELETE` en la tabla que considere apropiada.
7. (*Verificación IV*) Intente modificar el nombre del equipo 'Argentina' por 'ARG' utilizando un *script* de `UPDATE`. ¿Es posible hacerlo?
8. (*Actualización en cascada*) Las autoridades de la FIFA quieren que sea posible cambiar el nombre de los equipos por sus diminutivos, modificando automáticamente todas las filas que hacen referencia a ella en otras tablas. Modifique para ello el *script* de `CREATE TABLE`, definiendo una `CONSTRAINT` de `ON UPDATE` en la tabla correspondiente.
9. (*Verificación V*) Intente nuevamente modificar el nombre del equipo 'Argentina' por 'ARG' y analice lo ocurrido.
10. (*Eliminación en cascada*) Las autoridades de la FIFA insertaron a 'CHILE' por error y necesitan eliminar todos los registros que contengan este equipo. Necesitan poder el automáticamente eliminar todas las filas que hacen referencia a este equipo en otras tablas. Modifique para ello el *script* de `CREATE TABLE`, definiendo una `CONSTRAINT` de `ON DELETE` en la tabla correspondiente y analice lo ocurrido.
11. (*Verificación VI*) Intente nuevamente eliminar el equipo erróneamente agregado y analice lo ocurrido. ¿Qué ocurre si nos olvidamos el `WHERE` en un `DELETE`?

## ANEXO

Le sugerimos leer las siguientes entradas de la documentación de *PostgreSQL*, y en particular los ejemplos que las mismas incluyen:

- `CREATE TABLE`
- `PRIMARY KEY`
- `CONSTRAINT`
- `FOREIGN KEY, REFERENCES, ON UPDATE y ON DELETE`