



Tecnicatura Universitaria
en Desarrollo de Aplicaciones Informáticas

Base de Datos

Docentes

- Viviana Ferraggine
- Gustavo Adolfo Correa Reina
- Hernán Kulsgaard
- Sebastián Villar
- Cecilia Campos Lozia

Trabajo Práctico Especial

GRUPO N° 05

Integrantes

- ☐ Gándara, Matías
- ☐ López Ferreira, Javier

Informe

Introducción

El presente documento es un informe descriptivo de las tareas realizadas por dos estudiantes en el marco del Trabajo Práctico Especial de la asignatura Base de Datos perteneciente a la carrera Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires.

El referido Trabajo Práctico Especial consiste en la resolución de un conjunto de controles y servicios sobre una base de datos que mantiene un sistema de publicidad en línea. Este informe hace referencia a las tareas realizadas por este grupo de estudiantes que tuvieron el objetivo principal de dar respuesta a las consignas propuestas por los docentes.

Elaboración de Restricciones

Restricción	Tabla/s	Atributo/s	Tipo de restricción
a)	EVENTO_EDICION	fecha_inicio_pub; fecha_fin_pub	TUPLA
b)	SUBCATEGORIA	id_categoria count(*)	TABLA
c)	EVENTO_EDICION; PATROCINIOS	presupuesto; sum(aporte)	ASSERTION
d)	PATROCINANTE; PATROCINIOS; EVENTO_EDICION; EVENTO	id_distrito; id_patrocinante; id_evento; nro_edicion	ASSERTION

- a. *Se debe consistir que la fecha de inicio de la publicación de la edición sea anterior a la fecha de fin de la publicación del mismo si esta última no es nula.*

```
ALTER TABLE gr05_evento_edicion
ADD CONSTRAINT ck_inicio_fin_publicacion
CHECK (
    ( fecha_fin_pub IS NOT NULL
      AND fecha_fin_pub > fecha_inicio_pub )
    OR fecha_fin_pub IS NULL);
```

- b. *Cada categoría no debe superar las 50 subcategorías*

Restricción en SQL estándar declarativo:

```
ALTER TABLE gr05_subcategoria
ADD CONSTRAINT ck_maximo_subcategorias_x_categoria
CHECK NOT EXISTS
    (SELECT 1
     FROM gr05_subcategoria
     GROUP BY (id_categoria)
     HAVING COUNT(id_subcategoria) > 50);
```

- c. *La suma de los aportes que recibe una edición de un evento de sus patrocinantes no puede superar el presupuesto establecido para la misma.*

Restricción en SQL estándar declarativo:

```
CREATE ASSERTION ASS_APORTES_EXCEDEN_PRESUPUESTO
CHECK NOT EXISTS (
    SELECT 1
    FROM gr05_patrocinios p
    JOIN gr05_evento_edicion e
        ON (p.id_evento = e.id_evento
           AND p.nro_edicion = e.nro_edicion)
    GROUP BY p.id_evento, p.nro_edicion, e.presupuesto
    HAVING SUM(p.aporte) > e.presupuesto);
```

Tabla\ Evento	INSERT	UPDATE	DELETE
Patrocinios	Sí	aporte	No
Evento_Edicion	No	presupuesto	No

d. Los patrocinantes solo pueden patrocinar ediciones de eventos de su mismo distrito

Tabla\ Evento	INSERT	UPDATE	DELETE
Patrocinante	No	id_distrito	No
Patrocinios	Sí	id_patrocinante id_evento nro_edicion	No
Evento_Edicion	No	id_evento nro_edicion	No
Evento	No	id_distrito	No

Restricción en SQL estándar declarativo:

```
CREATE ASSERTION ASS_MISMO_DISTRITO_PATROCINIO_EVENTO
CHECK NOT EXISTS (
    SELECT 1
    FROM gr05_patrocinante p JOIN gr05_patrocinios pa
    ON p.id_patrocinante = pa.id_patrocinante JOIN gr05_evento_edicion e
    ON pa.id_evento = e.id_evento JOIN gr05_evento ev
    ON e.id_evento = ev.id_evento
    WHERE p.id_distrito = ev.id_distrito);
```

Definición de Vistas

1. *Identificador de los Eventos cuya fecha de realización de su último encuentro esté en el primer trimestre de 2020.*

```
CREATE VIEW v_eventos_trim_1_2020 AS
    SELECT id_evento
    FROM gr05_evento_edicion
    WHERE fecha_edicion
        BETWEEN to_date('01/01/2020', 'DD/MM/YYYY')
        AND to_date('31/03/2020', 'DD/MM/YYYY')
WITH LOCAL CHECK OPTION;
```

La vista no es actualizable ya que no conserva todas las columnas de la clave primaria. En este caso, falta la columna nro_evento de la tabla gr05_evento_edicion.

2. *Datos completos de los distritos indicando la cantidad de eventos en cada uno.*

--En SQL Standard

```
CREATE VIEW cant_eventos_distrito AS
    SELECT d.*, count(*) AS cant_eventos
    FROM gr05_distrito d
        JOIN gr05_evento e ON (e.id_distrito = d.id_distrito)
    GROUP BY d.id_distrito;
```

La vista es actualizable sobre la tabla distrito porque preserva la clave de esa tabla. Además, si bien existe una función de agregación (que viola uno de los requisitos de actualizable para vistas definidas sobre una tabla), se satisface la condición de que una fila dada en una tabla aparece como máximo una vez en la vista, para vistas definidas sobre más de una tabla. No obstante, la implementación en PostgreSQL no es actualizable, ya que impone que para que una vista sea actualizable debe haber una sola tabla en el FROM.

3. *Datos Categorías que poseen eventos en todas sus subcategorías.*

```
CREATE VIEW categoria_all_subcategorias_con_eventos AS
    SELECT *
    FROM gr05_categoria c
    WHERE NOT EXISTS(
        SELECT 1
        FROM gr05_subcategoria s
```

```
WHERE (c.id_categoria = s.id_categoria)
AND NOT EXISTS(
    SELECT id_categoria, id_subcategoria
    FROM gr05_evento e
    WHERE (s.id_categoria = e.id_categoria
        AND s.id_subcategoria = e.id_subcategoria)
)
);
WITH LOCAL CHECK OPTION;
```

La vista es actualizable sobre la tabla categoría porque preserva la clave de esa tabla. Además se satisface si una fila dada en una tabla aparece como máximo una vez en la vista. La implementación en PostgreSQL es actualizable, ya que se cumple con lo que exige PostgreSQL que para que una vista sea actualizable debe haber una sola tabla en el FROM.