

## Vistas

### Objetivos de Práctica de Vistas:

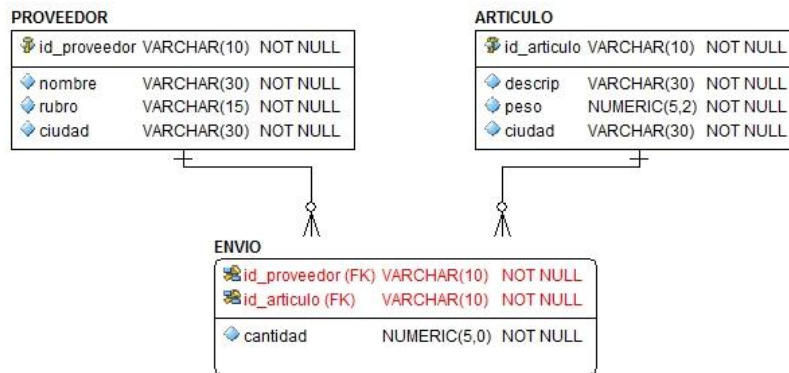
- Manejo de vistas: definición y actualización
- Actualizaciones en las vistas provenientes de sus componentes
- Vistas actualizables: definición, Key Preserved, triggers INSTEAD OF

Nota1: La cláusula WITH CHECK OPTION está disponible en PostgreSQL a partir de la Versión 9.4

Nota2: Recuerde que no tiene privilegios para crear Vistas o Triggers sobre los esquemas de la materia (unc\_esq\_peliculas.\*, unc\_voluntario.\*) Para poder realizar los ejercicios en PostgreSQL debe crear las tablas en su propio esquema con la sentencia CREATE TABLE AS SELECT

### Ejercicio 1

Considerando el siguiente DERExt



a) definir las siguientes Vistas:

- *ENVIOS500* con los envíos de más de 500 unidades de algún artículo(a partir de ENVIO) **desarrollado en clase**
- *PRODUCTOS\_MAS\_PEDIDOS* con los diferentes artículos que han tenido al menos un envío de más de 500 unidades (a partir de ENVIOS500)
- *ENVIOS\_PROV* con los diferentes id de proveedor y la cantidad total de unidades enviadas(a partir de Envío)
- *DETALLE\_ENVIOS* que contenga, para cada envío de más de 500 unidades, la descripción y el peso del artículo, el nombre del proveedor y la cantidad enviada (a partir de Proveedor, Artículo, ENVIOS500)

b) Determinar si las Vistas anteriores son actualizables o no (en este caso indicar la causa)

c) Determinar el efecto de las siguientes operaciones sobre la Vista *Envios500*, considerando si tiene especificada opción de chequeo (WITH CHECK OPTION) o no (**parcialmente desarrollado en clase**):

- INSERT INTO *ENVIOS500* VALUES ('P1', 'A1', 500);
- INSERT INTO *ENVIOS500* VALUES ('P2', 'A2', 300);
- UPDATE *ENVIOS500* SET cantidad=1000 WHERE id\_proveedor='P1';
- UPDATE *ENVIOS500* SET cantidad=100 WHERE id\_proveedor='P1';

### Ejercicio 2

Considere las siguientes sentencias de creación de vistas, discuta si las mismas son actualizables o no y justifique.

- CREATE VIEW Distribuidor\_200 AS  
SELECT id\_distribuidor, nombre, tipo  
FROM unc\_esq\_peliculas.distribuidor  
WHERE id\_distribuidor > 200;
- CREATE VIEW Departamento\_dist\_200 AS  
SELECT id\_departamento, nombre\_departamento, id\_ciudad, jefe\_departamento  
FROM unc\_esq\_peliculas.departamento d

**Vistas**

WHERE id\_distribuidor > 200;

c. Algunos registros de la tabla Distribuidor son:

id_distribuidor	nombre	direccion	telefono	tipo
1049	Distribuidor 1049	Doro	7372214-6352	N
1050	Distribuidor 1050	Lakhagarh	569842-2643	N

Y sea la vista:

```
CREATE VIEW Distribuidor_1000 AS  
SELECT *  
FROM unc_esq_peliculas.distribuidor d  
WHERE id_distribuidor > 1000;
```

¿Si se ejecuta? la siguiente sentencia

INSERT INTO Distribuidor\_1000 **VALUES** (1050, NuevoDistribuidor 1050, “Montiel 340”, “569842-2643”, N);

Indique y justifique la opción correcta:

- I. Falla porque la vista no es actualizable;
- II. Falla porque si bien la vista es actualizable viola una restricción de integridad referencial.
- III. Falla porque si bien la vista es actualizable viola una restricción de integridad de unicidad.
- IV. Procede exitosamente

**Ejercicio 3**

Considere el esquema de la BD unc\_esq\_peliculas:

- a) Escriba las sentencias de creación de cada una de las vistas solicitadas en cada caso.
  - b) Indique si para el estándar SQL dicha vista es actualizable o no, si es de Proyección-Selección o Proyección-Selección-Ensamble. Justifique cada respuesta.
1. Cree una vista EMPLEADO\_DIST\_20 que liste el id\_empleado, nombre, apellido, sueldo y fecha\_nacimiento de los empleados que pertenecen al distribuidor cuyo identificador es 20.
  2. Sobre la vista anterior defina otra vista EMPLEADO\_DIST\_2000 con el nombre, apellido y sueldo de los empleados que cobran más de 2000.
  3. Sobre la vista EMPLEADO\_DIST\_20 cree la vista EMPLEADO\_DIST\_20\_70 con aquellos empleados que han nacido en la década del 70 (entre los años 1970 y 1979).  

Analice cuales serían los controles y el comportamiento ante actualizaciones sobre las vistas EMPLEADO\_DIST, EMPLEADO\_DIST\_2000 y EMPLEADO\_DIST\_20\_70 si están definidas con WITH CHECK OPTION LOCAL o CASCADE en cada una de ellas. Evalúe todas las alternativas.
  4. Cree una vista PELICULAS\_ENTREGADAS que contenga el código de cada película y la cantidad de unidades entregadas.
  5. Cree una vista DISTRIBUIDORAS\_ARGENTINA con los datos completos de las distribuidoras nacionales (incluya nro\_incripcion y encargado)
  6. Usando la vista anterior, cree la vista DISTRIBUIDORAS\_MAS\_2\_EMP con los datos completos de las distribuidoras nacionales cuyos departamentos tengan más de 2 empleados.

### Vistas

#### **Ejercicio 4**

Para las siguientes vistas de ensamble definidas debajo indique:

- a) A qué tipo de vista, según la ubicación de la FK, corresponde cada una? (Pág. 18 DBI\_2016\_09\_Vistas.pdf)
  - b) Indique cual es la clave preservada en cada caso y de qué tabla proviene.
  - c) ¿Qué atributos se pueden actualizar en cada caso?
  - d) Escriba los triggers INSTEAD OF en Postgresql para actualizar cada una de las vistas.
- 
1. CREATE VIEW NACIONAL\_KP\_1 AS  
SELECT N.Id\_distribuidor, nro\_inscripcion, id\_distrib\_mayorista, encargado,  
nombre, direccion, telefono, tipo  
FROM unc\_esq\_peliculas.distribuidor D NATURAL JOIN unc\_esq\_peliculas.nacional N;
  
  2. CREATE VIEW CIUDAD\_KP\_2 AS  
SELECT id\_ciudad, nombre\_ciudad, C.id\_pais, nombre\_pais  
FROM unc\_esq\_peliculas.ciudad C NATURAL JOIN unc\_esq\_peliculas.pais P;
  
  3. CREATE VIEW ENTREGAS\_KP\_3 AS  
SELECT nro\_entrega, re.codigo\_pelicula, cantidad, titulo  
FROM unc\_esq\_peliculas.renglon\_entrega RE NATURAL JOIN unc\_esq\_peliculas.pelicula P;