

---

## INTRODUCCIÓN

El Trabajo Práctico Especial consiste en la resolución de un conjunto de controles y servicios sobre una base de datos que representa a alguno de los modelos que se dan en el Anexo I. Las consignas para su desarrollo se indican en la sección “**Pautas de Desarrollo**” y el **esquema de la base de datos** se encuentra disponible en formato SQL exportado desde Vertabelo.

## TAREAS A DESARROLLAR

NOTA: en todos los casos se debe reemplazar a **XX** por el *número de grupo* asignado.

### A. AJUSTE DEL ESQUEMA

- A partir del script de SQL (E\*\_create.sql) de creación de cada esquema debe ajustar dicho esquema según las **pautas de nomenclatura** indicada en la sección [REGLAS DE NOMBRES](#).
- El script de generación del esquema se debe llamar **GXX\_creacion.sql**. (ejemplo: el grupo 14 generará el archivo G14\_creacion.sql)

### B. SERVICIOS

Implemente lo siguiente con el recurso que crea más conveniente (triggers, procedimientos y/o vistas). Si necesita para este ítem crear tablas adicionales al esquema creado en el script de creación (GXX\_Creacion.sql) incorpórelas en el script **GXX\_Cambios.sql**.

### C. DEFINICIÓN DE VISTAS

Escriba la sentencia SQL para crear las vistas (En lo posible actualizables en Postgresql) e incorpórelas en el script **GXX\_Cambios.sql**. En cada caso comente y ejemplifique en el script si es actualizable o no, indicando la/s causas/s.

## PAUTAS DE ENTREGA

### Consideraciones Importantes

- Se recomienda leer el enunciado completo del trabajo antes de resolverlo
- Se deberán respetar
  - las pautas de entrega indicadas en cada parte del enunciado y
  - las pautas de nomenclatura para todos los objetos del esquema de la base de datos, indicadas en la sección Reglas de Nombres

- Los scripts entregados se probarán y testearán en **PostgreSQL V.11 utilizando DataGrip**. Cada grupo debe asegurarse que los scripts ejecuten completos sin errores para que la cátedra pueda proceder a la corrección del TPE, en el orden establecido; por esto es recomendable que los chequee debidamente a fin de asegurar que los mismos se ejecuten sin inconvenientes.
- Todo el **código SQL entregado debe estar optimizado**, es decir utilizando estrictamente los atributos, tablas y/u operaciones necesarias, de acuerdo a lo requerido.

**Nota:** Se descontarán hasta 2 puntos de la nota final por no respetar las pautas de entrega, reglas de nombres solicitadas y/o que se deban re-entregar los script por tener errores al momento de testearlos.

## ENTREGA DEL TRABAJO

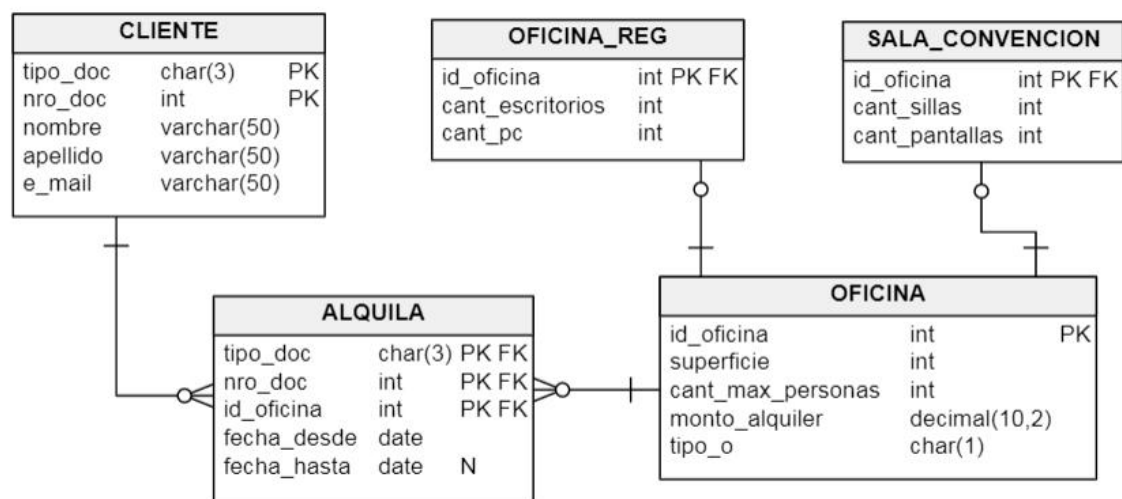
Se deberá completar una tarea (ENTREGA) en la plataforma Moodle en la que se deberán subir los **archivos scripts solicitados** en forma separada. En la misma tarea se confirmará el día siguiente a la entrega, si los scripts corrieron sin errores o, de haber advertido algún inconveniente en su ejecución, se informará el primer error encontrado.

Se deberán entregar 3 (tres) scripts, que se deben poder ejecutar en orden (detallados en los ítems 1, 2 y 3) y sin producir errores:

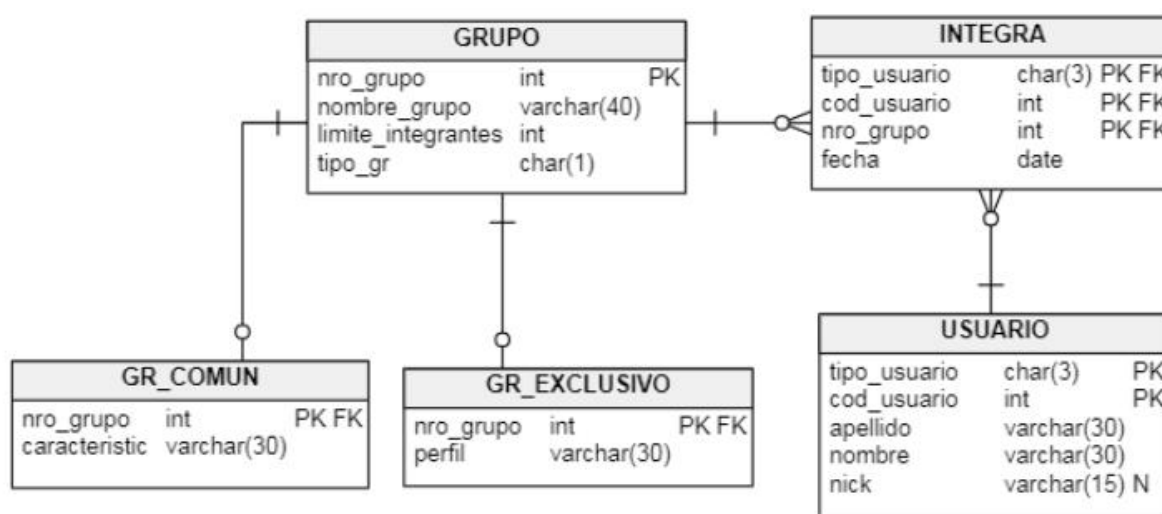
1. script GXX\_Creación.sql
2. script GXX\_Cambios.sql
3. script GXX\_Borrado.sql

## ESQUEMAS

## ESQUEMA E1



## ESQUEMA E2



## REGLAS DE NOMBRES

Para el desarrollo y corrección de los esquemas se seguirán estos estándares que permiten identificar cada trabajo y objeto con nombres completos y descriptivos.

### *Consideraciones Generales*

El prefijo de grupo **<GRXX>** se usará en tablas, vistas, procedimientos almacenados, triggers, índices y constraints. XX se deberá reemplazar por el número de grupo asignado por la cátedra en cada caso. La nomenclatura a seguir es:

1. Tablas/Vistas: **<GRXX>\_<nombre\_tabla o vista>**
2. Índices: **IDX\_<GRXX>\_<nombre\_indice>**
3. Constraints

- **Clave Primaria: PK\_<GRXX>\_<nnnn>**

donde: nnnn = el nombre de la tabla para la que la constraint se construye (Por ejemplo: PK\_GRXX\_Beneficiario)

- **Clave Extranjera: FK\_<GRXX>\_<pppp>\_<cccc>**

donde pppp = el nombre de la tabla referenciante u original, cccc = el nombre de la tabla referenciada.

- **Unique: UQ\_<GRXX>\_<nnnn>\_<cccc>**

donde: nnnn = el nombre de la tabla para la que la constraint se construye, cccc = el nombre del campo interviniente.

4. Procedimientos, Triggers y Funciones

- **Triggers: TR\_<GRXX>\_<tbl>\_<nnnn>**

siendo <nnnn> el nombre que considere adecuado para el trigger y <tbl> la tabla a la cual está asociada el trigger.

- **Procedimientos: PR\_<GRXX>\_<nnnn>**

siendo <nnnn> el nombre que considere adecuado para el procedimiento.

- **Funciones: FN\_<GRXX>\_<nnnn>**

siendo <nnnn> el nombre que considere adecuado para la función.

- **Funciones de triggers: TRFN\_<GRXX>\_<nnnn>**

siendo <nnnn> el nombre que considere adecuado para la función de trigger.

**RECORDAR:**

**al momento de realizar los script que:**



Aquellos insert, update o delete que provocan fallos de por alguna restricción, es decir los que son de test de restricciones deben colocarse como comentarios /\* \*/.

Recuerde colocarle CASCADE a las sentencias drop table si realiza el borrado de las tablas en cualquier orden, sino tiene que respetar el orden dado por las foreign key.

Una función que es llamada por un trigger debe estar dentro del script siempre antes del create trigger que la invoca.

Recuerde que aquellas sentencias que Postgresql no soporta deben ir comentariadas /\* \*/

**ANEXO I - DESARROLLO*****Para Esquema E1:*****A. TRIGGERS Y SERVICIOS**

1. Se le agrega columna cantidad\_salas (cantidad de salas de convención) y cantidad\_oficinas (cantidad de oficinas regulares) a la tabla CLIENTE. Es necesario mantener actualizadas las columnas cantidad\_salas y cantidad\_oficinas con la cantidad de salas de convención y oficinas regulares que cada cliente tiene alquiladas por tiempo indeterminado (es decir que aún no tienen fecha de fin). Se debe realizar con triggers FOR STATEMENT.
2. Utilizando 2 vistas V\_OFICINA\_REGULAR y V\_SALA\_CONVENCION que contienen todos los datos de las oficinas regulares o de las salas de convención respectivamente, construir los triggers INSTEAD OF necesarios para mantener actualizadas las tablas de OFICINA, OFICINA\_REG y SALA\_CONVENSION de manera de respetar el diseño de datos de la jerarquía.

**B. VISTAS**

1. Construya una vista V\_CLIENTES\_COMP que contenga las oficinas que han sido alquiladas por todos los clientes.
2. Construya una vista V\_OFICINAS\_REG que liste para cada oficina su identificador, su tipo, su superficie, su monto de alquiler y la cantidad promedio de escritorios por superficie.

***Para Esquema E2:*****A. TRIGGERS Y SERVICIOS**

1. Se le agrega la columna cantidad\_grupos\_comun (cantidad de grupos comunes) y cantidad\_grupos\_excl (cantidad de grupos exclusivos) a la tabla USUARIO y la columna activo a la tabla INTEGRA. Es necesario mantener actualizada las columnas cantidad\_grupos\_comun y cantidad\_grupos\_excl con la cantidad de grupos comunes y excluivos activos que pertenece cada usuario (es decir que el campo activo está en true). Se debe realizar con triggers FOR STATEMENT.
2. Utilizando 2 vistas V\_GRUPO\_COMUN y V\_GRUPO\_EXCLUSIVO que contienen todos los datos de los grupos comunes o exclusivos respectivamente, construir los triggers INSTEAD OF necesarios para mantener actualizadas las tablas de GRUPO, GR\_COMUN y GR\_EXCLUSIVO de manera de respetar el diseño de datos de la jerarquía.

**B. VISTAS**

1. Construya una vista V\_GRUPOS\_COMP que contenga los grupos que son integrados por todos los usuarios.
2. Construya una vista V\_OFICINAS\_REG que liste para cada grupo la cantidad de usuarios que integran cada grupo y el promedio su identificador, su tipo, su superficie, su monto de alquiler y la cantidad promedio de escritorios por superficie.