

Fundamentos de Bases de Datos.

Práctica 5.

Profesora: Dra. Amparo López Gaona
alg@ciencias.unam.mx

Laboratorio: Lic. Carlos Augusto Escalona Navarro
caen@ciencias.unam.mx

21 de octubre de 2021

Se dan a conocer especificaciones de entrega para la práctica 5.

1. DDL

Un esquema de base de datos se especifica mediante un conjunto de definiciones expresadas mediante un lenguaje especial llamado **lenguaje de definición de datos (DDL)**¹.

Un diccionario de datos contiene metadatos, es decir, información acerca de los objetos que se encuentran almacenando dentro de la base de datos. El esquema es el espacio de almacenamiento físico que contiene desde la definición de una tabla hasta los disparadores además la información que sean necesarios para la administración de la base de datos. Un sistema manejador de base de datos (**SMBD**), consulta el diccionario de datos (metadatos) para verificar que sea una operación válida o que este definido en éste, antes de leer o modificar los datos que ya se encuentran almacenados dentro de la base de datos.

Especificamos los métodos de acceso usados por el sistema manejador de bases de datos, por un conjunto de instrucciones de un tipo especial de DDL, denominado lenguaje de almacenamiento y definición de datos. Estas instrucciones definen los detalles de implementación de los esquemas de base de datos, que se ocultan usualmente a los usuarios de los esquemas y esta información queda visible para los usuarios administradores de la base de datos.

¹Siglas en inglés de **Data Definition Language**

2. pgAdmin 4

PgAdmin 4 es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados (EnterpriseDB Postgres Plus Advanced Server y Greenplum Database). Incluye:

- Interfaz administrativa gráfica.
- Herramienta de consulta SQL (con un EXPLAIN gráfico).
- Editor de código procedural.
- Agente de planificación SQL/shell/batch.
- Administración de Slony-I.

PgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas.

La interface gráfica soporta todas las características de PostgreSQL y **hace simple la administración**. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows , Linux, FreeBSD, Mac OSX y Solaris.

PgAdmin 4 versión 4.11. Esta versión de pgAdmin 4 incluye más de 34 correcciones de errores y nuevas características. Los cambios notables en esta versión incluyen:

- Se agrego soporte para la edición de conjuntos de resultados en la Herramienta de consulta, si los datos pueden identificarse como actualizables.
- Se agrego las opciones EXPLICAR para CONFIGURACIÓN y RESUMEN.
- Se estable el cursor del mouse adecuadamente según el estado de bloqueo del diseño.
- Evita el parpadeo de la información sobre herramientas de gran tamaño en el lienzo de EXPLICACIÓN gráfica.
- Se corrigió un error al recuperar datos json de la tabla.
- Se corrigió el error 'literal de matriz mal formado' al actualizar un trabajo pgAgent.
- Se corrigió un error que se podía ver al editar privilegios de columna.
- Se soluciona un error visto al actualizar una cadena de conexión en un paso de trabajo pgAgent.
- Las opciones de EXPLAIN deben ser específicas de la instancia de la herramienta de consulta.
- Se permite la navegación por teclado de todos los controles en cuadrículas de subnodos.

3. Instrucciones DDL

Instrucción	Función
CREATE	se usa para crear una base de datos, tabla, vistas , etc.
ALTER	se utiliza para modificar la estructura, por ejemplo añadir o borrar ocolumnas de una tabla.
DROP	con esta sentencia, podemos eliminar los objetos de la estructura, o por ejemplo un índice o una secuencia.

3.1. Crear tablas

Crea una nueva tabla en Postgres:

```
CREATE TABLE table_name (  
[ { column_name data_type [ column_constraint [ ... ] ] } ] ;  
  
where column_constraint is:  
  
[ CONSTRAINT constraint_name ]  
{ NOT NULL |  
  NULL |  
  CHECK ( expression ) |  
  DEFAULT default_expr |  
  UNIQUE index_parameters |  
  PRIMARY KEY index_parameters |  
  REFERENCES reftable [ ( refcolumn ) ] [ MATCH FULL | MATCH PARTIAL  
  | MATCH SIMPLE ] [ ON DELETE action ] [ ON UPDATE action ] }  
[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
```

table_name: Es el nombre de la nueva tabla. Los nombres de tabla deben seguir las reglas para identificadores. table_name puede tener **un máximo de 128 caracteres**, excepto para los nombres de tablas temporales locales (nombres precedidos de un solo signo de número (#)) que no puede **superar los 116 caracteres**.

3.2. Editar tablas

Modifica una definición de tabla al alterar, agregar o quitar columnas y restricciones, o deshabilitar o habilitar restricciones y disparadores.

```
ALTER TABLE name [ * ]  
  action [, ... ] ;
```

```
ALTER TABLE name [ * ]
RENAME [ COLUMN ] column TO new_column;
```

```
ALTER TABLE name RENAME TO new_name;
```

```
ALTER TABLE name SET SCHEMA new_schema;
```

where action is one of:

```
ADD [ COLUMN ] column data_type [ COLLATE collation ]
[ column_constraint [ ... ] ]
DROP [ COLUMN ] [ IF EXISTS ] column [ RESTRICT | CASCADE ]
ALTER [ COLUMN ] column [ SET DATA ] TYPE data_type
[ COLLATE collation ] [ USING expression ]
ALTER [ COLUMN ] column SET DEFAULT expression
ALTER [ COLUMN ] column DROP DEFAULT
ALTER [ COLUMN ] column { SET | DROP } NOT NULL
ALTER [ COLUMN ] column SET STATISTICS integer
ALTER [ COLUMN ] column SET ( attribute_option = value [, ... ] )
ALTER [ COLUMN ] column RESET ( attribute_option [, ... ] )
ALTER [ COLUMN ] column SET STORAGE { PLAIN | EXTERNAL | EXTENDED | MAIN }
ADD table_constraint [ NOT VALID ]
ADD table_constraint_using_index
DROP CONSTRAINT [ IF EXISTS ] constraint_name [ RESTRICT | CASCADE ]
```

3.3. Comentarios sobre tablas y columnas

Por buenas practicas de diseño de Bases de Datos es importante comentar las tablas y sus atributos para saber mas acerca de por que decidimos definir los objetos de la base de esa manera.

```
COMMENT ON
{
    COLUMN relation_name.column_name |
    CONSTRAINT constraint_name ON table_name |
    DATABASE object_name |
    DOMAIN object_name |
    FUNCTION function_name ( [ [ argmode ] [ argname ] argtype [, ...] ] ) |
    [ PROCEDURAL ] LANGUAGE object_name |
    SCHEMA object_name |
    SEQUENCE object_name |
    TABLE object_name |
} IS 'text'
```

4. Actividad

Deberán crear las instrucciones DDL para crear **las tablas** de bases de datos utilizando como base el diagrama **relacional** que realizaron en la práctica pasada.

Deben especificar las **llaves foráneas, candidatas y primarias** de cada tabla, recuerda que estas llaves pueden ser compuestas o simples; así como las restricciones con las que puede contar la tabla, no olvides **agregar los comentarios a cada una de tus tablas y columnas** de estas.

5. Entregables

Se debe crear un script DDL.sql el cual contendrá las instrucciones para la creación de la base de datos, sus tablas y sus restricciones, este archivo se encontrará en la carpeta **SQL**.