



Grupo: 16

Integrantes: Sole-Rene-Serruya

Correo: daianrene99@gmail.com

Objetivos

En el trabajo práctico 1 se crearon los modelos físicos de datos y las estructuras de datos para disponer de una base de datos única desde la que se nutrirá de información para la gestión de acciones para combatir la pandemia de COVID-19.

Los objetivos de este segundo práctico serán:

- Incorporar datos geográficos, de casos y de determinaciones PCR (testeos) a la base de datos creada en el primer práctico;
- explotar la información cargada con la realización de consultas;
- codificar *triggers* para preservar y auditar cambios en la estructura de datos.

Ejercicio 1: Unidades territoriales

Incorporar al esquema de *unidades territoriales* creado en el TP1 la información de provincias, departamentos y localidades dispuestas en:

<https://datos.gob.ar/ar/dataset/jgm-servicio-normalizacion-datos-geograficos>

En este sitio, elegir y descargar los archivos con formato “csv”:

- provincias.csv
- departamentos.csv
- localidades-censales.csv

Los mismos deben ser descargados en una carpeta de trabajo para luego importarlos en la base de datos.

En el caso de haber elegido PostgreSQL utilizar el comando “*copy*”; en el caso de SQLServer utilizar la clausula adicional “*From OpenRowSet Bulk ...*” .

Se recomienda que los datos importados desde el sistema operativo sean ubicados en tablas intermedias creadas en un esquema de trabajo (por ejemplo, “*importacion*”), con el objeto de verificar primero que todas las filas de los archivos sean importadas correctamente y luego, procesar las mismas para incorporarlas al modelo en las tablas definitivas.



Ejercicio 2: Determinaciones

Codificar funciones o procedimientos almacenados que permitan en forma dinámica importar archivos de determinaciones de casos COVID-19 descargados de <https://datos.gob.ar/dataset/salud-covid-19-determinaciones-registradas-republica-argentina>

La función (o procedimiento) debe recibir como argumentos la ubicación del archivo y el nombre de archivo.

La función debe retornar string que informe el resultado de la ejecución de la misma y la cantidad de filas importadas.

Se recomienda que los datos sean inicialmente importados en una tabla temporaria, para distinguir con consultas sobre la misma las localidades referenciadas. Si alguna no existe, la función debe informar la misma y se debe retornar sin realizar ninguna inserción.

Los datos deben ser incorporados en la tabla de determinaciones creada en el TP1. Por cada fila a importar debe verificarse previamente su existencia, ya que solo deben registrarse las filas que no existen, suponiendo una incorporación diaria y correlativa de los archivos. De haber registros con fecha anterior al 1/1/2020, no deben incorporarse a la tabla definitiva.

La función no debe manejar transacciones, pero debe considerarse que se la podrá invocar de la siguiente forma:

```
begin transaction;  
select f_importar_determinaciones('/tmp','/tmp/DeterminacionesCovid19.csv');
```

Si todo salió bien el operador podrá decidir si realiza un “commit”; o dar un “rollback” si se presentó algún error.



Ejercicio 3: Casos

Codificar funciones o procedimientos almacenados que permitan en forma dinámica importar archivos de casos COVID-19 descargados de <https://datos.gob.ar/dataset/salud-covid-19-casos-registrados-republica-argentina>.

La función (o procedimiento) debe recibir como argumentos la ubicación del archivo y el nombre de archivo.

Debe retornar un *tipo de dato de usuario* que permita distinguir:

- *atributo entero indicando resultado de la ejecución: 0 sin errores; valor distinto de 0 para informar errores;*
 - *atributo varchar para describir resultado de la ejecución: “Ejecución sin errores”, “Ejecución con errores”;*
 - *atributo entero mostrando cantidad de filas importadas;*
 - *atributo varchar para describir información faltante para continuar con la importación.*
- Por ejemplo:*

*Se recibió el valor “*sin dato*” en el campo Origen de financiamiento, y no está registrado previamente.*

Los datos deben ser incorporados en la tabla de casos creada en el TP1. Por cada archivo importado, se debe generar una fila en la tabla actualización y por cada fila del archivo importado, una fila en la tabla de casos. De haber registros con fecha anterior al 1/1/2020, no deben incorporarse a la tabla definitiva.

Se recomienda que los datos sean inicialmente importados en una tabla temporaria, para distinguir con consultas sobre la misma la siguiente información:

- distintos valores de provincia (carga y residencia)
- distintos valores de departamento
- país
- clasificación
- origen de financiamiento

Si alguna no existe, la función debe informar la misma y se debe retornar sin realizar ninguna inserción.

En lo que hace al manejo de transacciones, se hace la misma aclaración que para el caso de las determinaciones: la función no debe manejar transacciones, dando al operador la posibilidad de decidir si realizar un “commit” o “rollback” en función del resultado.



Ejercicio 4: Explotación de datos

Utilizando las funciones creadas en los ejercicios 2 y 3, importar como mínimo 3 días de archivos de determinaciones y 3 días de archivos de casos.

Usar los datos importados para dar respuesta a las siguientes estadísticas:

- Cantidad total de determinaciones realizadas por mes y por localidad, ordenadas de ese modo, informando total y positivos. Salida tipo tabla.
- Suponiendo que el valor de cada determinación tiene un costo de \$2800,- cual es el monto que lleva invertido el estado (Origen público de financiación) en el AMBA? Y cuanto en el resto del país? Salida tipo tabla, una única fila.
- Certeza de los test realizados: comparar los test realizados por cada provincia con los casos confirmados de esa provincia.

Provincia	Mes	Test realizados	Test positivos	Casos confirmados
Santa Fe	5	2500	1800	2000
	6	3500	3200	3150
Tucumán	6	1288	1200	1000
...				

Salida tipo tabla

Provincia: utilizar provincia de residencia para los casos.

Mostrar nombre de provincia, mes, cantidad total de tests realizados, cantidad total de test que dieron positivo y cantidad de casos confirmados para esa provincia y ese mes.

Ordenar por provincia y mes.

El script de solución propuesta para este ejercicio puede utilizar recursos propios de un motor de bases de datos en particular. Pero, en el caso de ser posible, se preferirán las soluciones simples, con uso de SQL estándar.



Ejercicio 5: Auditoría de cambios

Diseñar una estructura de auditoría que permita registrar los cambios de datos realizados en determinadas tablas por los usuarios. Las instrucciones cuyos cambios se desean auditar son “INSERT”, “UPDATE” y “DELETE”. Para detectar cambios deben utilizarse *triggers*.

Se debe contemplar el registro de los siguientes elementos como mínimo:

- identificador (id) de la fila afectada
- usuario que realizó el cambio
- momento
- tipo de instrucción DML
- Valor previo y valor nuevo

Las tablas a definir deben ubicarse en un esquema aparte, denominado “auditoría”. Por ejemplo, tabla “log_auditoria” en el esquema “auditoria”.

Programar las funciones/triggers necesarios para auditar cambios en las tablas de Unidades territoriales.

Opcional:

¿Qué pasa si se desea extender el uso del trigger a algunas otras tablas? Proponer alternativas de mejoras.

Recordar incluir en la entrega, tanto los scripts de creación de estructuras, funciones y triggers, como ejemplos utilizados en las pruebas.



MODALIDAD DE ENTREGA

- Los trabajos deben entregarse el martes 24 de Octubre.
- Los grupos deben ser los mismos que los del TP1.
- Solo participarán del TP2 los grupos que hayan entregado TP1.
- Deben remitirse por correo electrónico a: fich.informatica.bd@gmail.com
- En el asunto del correo se debe indicar “TP 2 – Grupo XX”
- Se debe adjuntar un único archivo comprimido (.zip o .rar) con el nombre “TP 2 – Grupo XX”, cuyo contenido sea:
 1. “TP2 – Grupo xx.pdf”: documento con indicaciones particulares o explicaciones para facilitar la corrección y validación del trabajo práctico. Incluir integrantes del grupo en este documento.
 2. “01-Script-Previo.sql”: toda instrucción que sea necesaria para hacer cambio de estructura en la base de datos creada con el TP1. Los scripts del TP2 se ejecutarán sobre la base de datos resultado del TP1, por lo tanto, deberán incluirse en este script todos los comandos que se consideren necesarios.
 3. “02-Importacion-UnidadesTerritoriales.sql”: scripts de importación de datos de provincias, departamentos y localidades.
 4. “03-Importacion-Casos.sql”: scripts de creación y ejecución de funciones de importación de datos de casos y determinaciones.
 5. “04-Consultas.sql”: scripts conteniendo las consultas solicitadas en el enunciado del TP2.
 6. “05-Triggers.sql”: scripts de creación de estructuras y funcionalidades necesarias para implementar el trigger de auditoría solicitado.