De la creación a la manipulación de una base de datos relacional

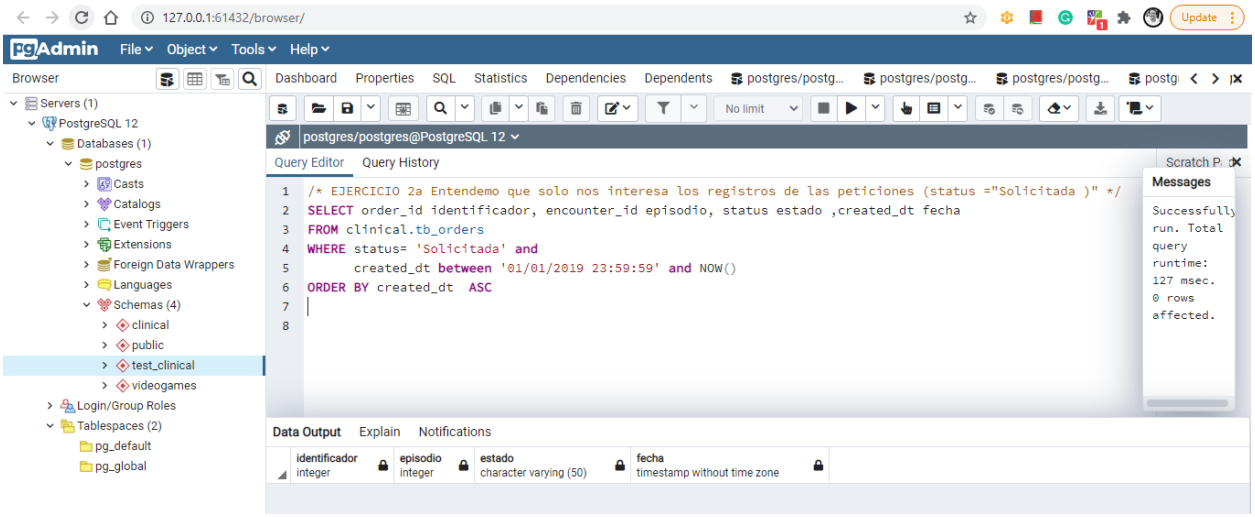
NOMBRE Y APELLIDOS: Jorge Oliva Ramos

EJERCICIO 1 (20%)

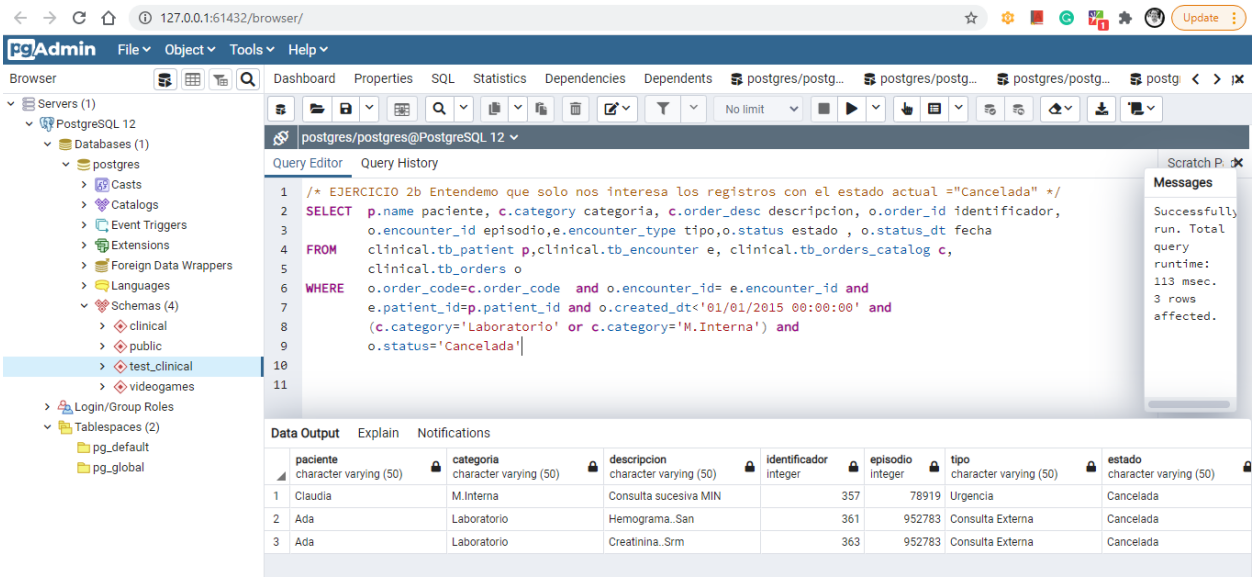
El resultado de este ejercicio se adjunta en el fichero : pec2_ej1.sql

EJERCICIO 2 (30%)

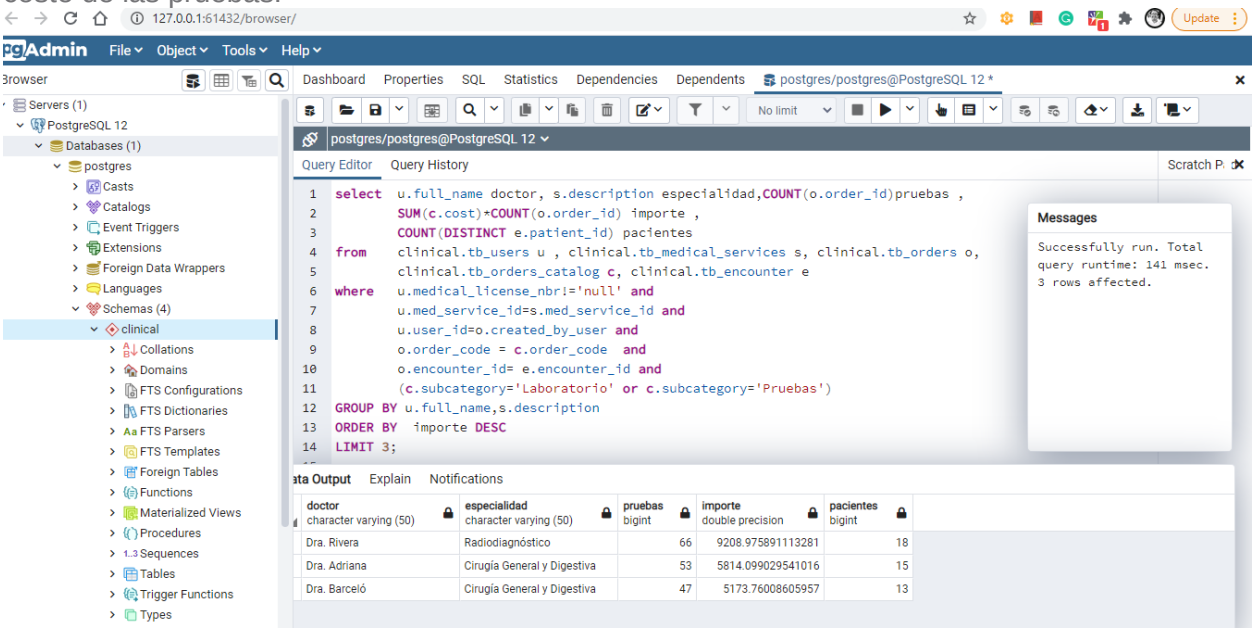
a) Proporcionar el identificador, el episodio, el estado y la fecha de creación de las peticiones de prestaciones, ordenadas ascendentemente, desde el 1 de enero de 2019 hasta la fecha de ejecución de la consulta.



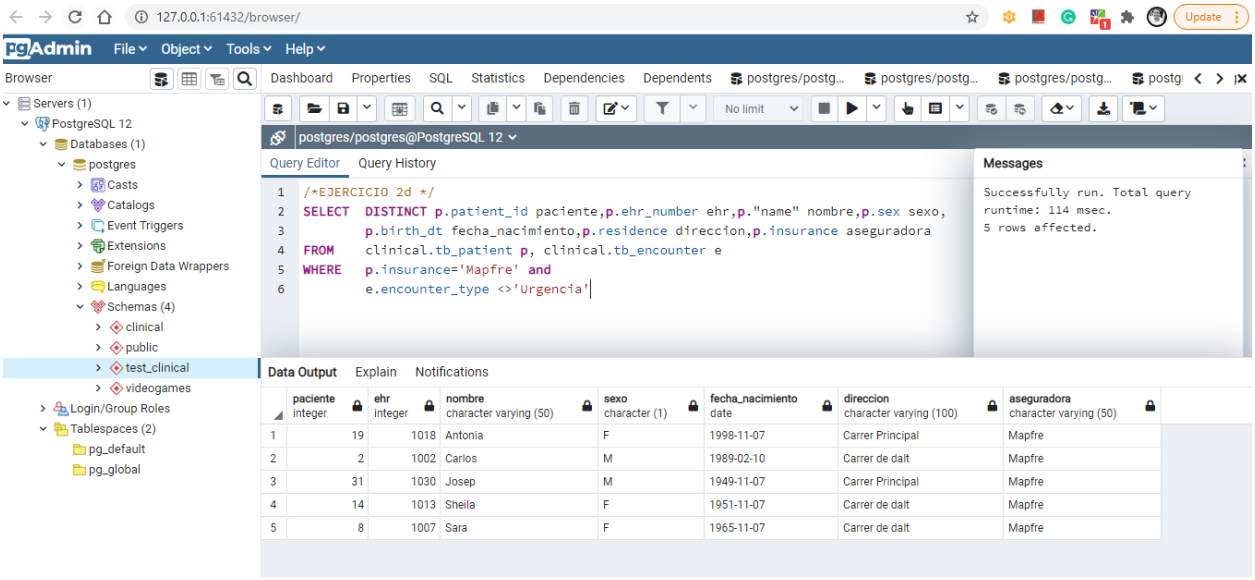
b) Proporcionar la lista de los pacientes a los que se les pidió una prueba de categoría Laboratorio o de M.Interna con anterioridad a 2015. y que finalmente fue cancelada. Concretamente queremos que aparezca: el nombre del paciente, la categoría de la prueba, la descripción, el identificador de la prestación, el tipo de episodio, el estado de la prestación y la fecha en que se canceló.



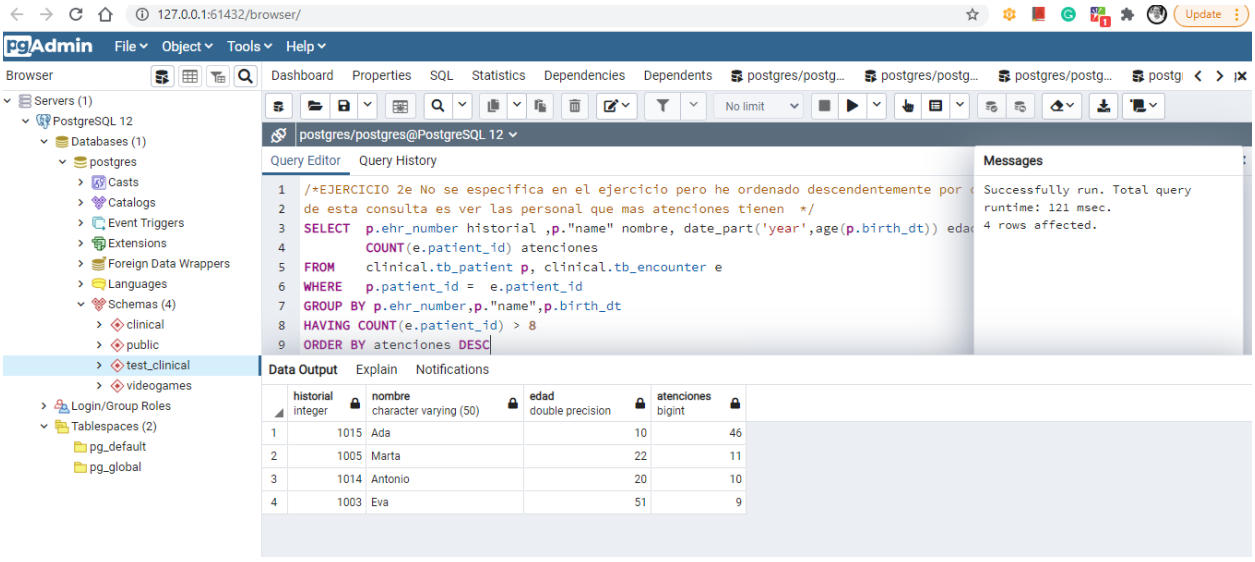
c) Proporcionar la lista de los **tres** facultativos que más han gastado en pruebas. Queremos su nombre y especialidad, junto al número de pruebas que ha pedido, el coste de todas ellas y el número de pacientes diferentes a que corresponden. Ordenados de mayor a menor por el coste de las pruebas.



d) Proporcionar la lista con todos los datos de los pacientes diferentes de la aseguradora Mapfre que no han sido atendidos nunca de manera urgente.

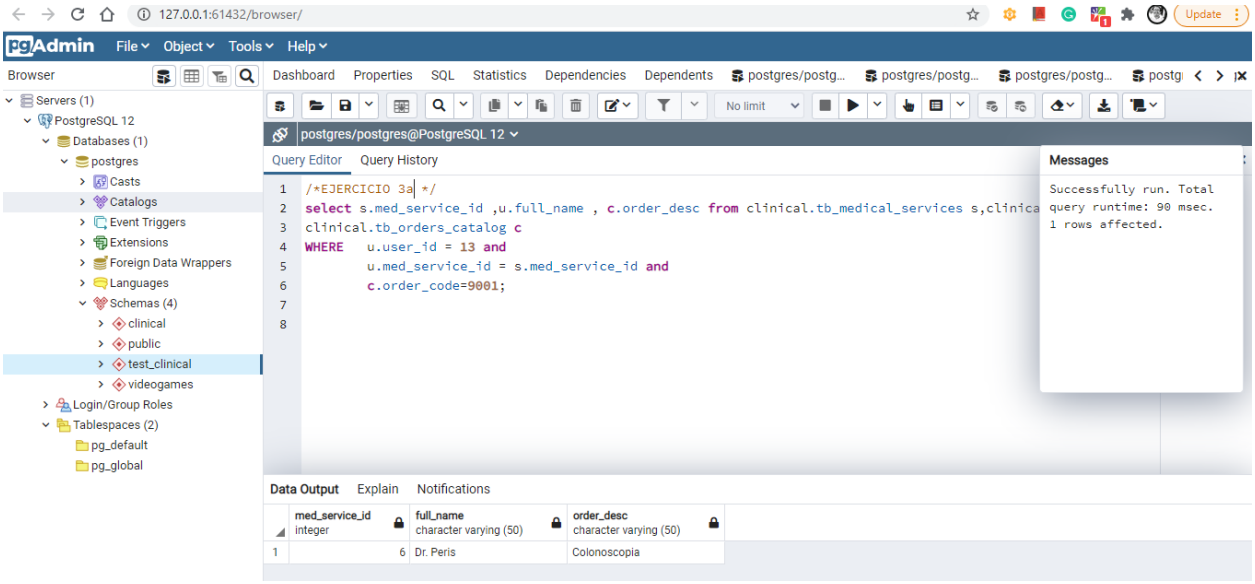


e) Proporcionar el número de historial clínico, el nombre, la edad de los pacientes y el número de veces que han sido atendidos en la clínica, de aquellos a los que se ha atendido más de 8 veces, ordenados por el número de veces que han sido atendidos

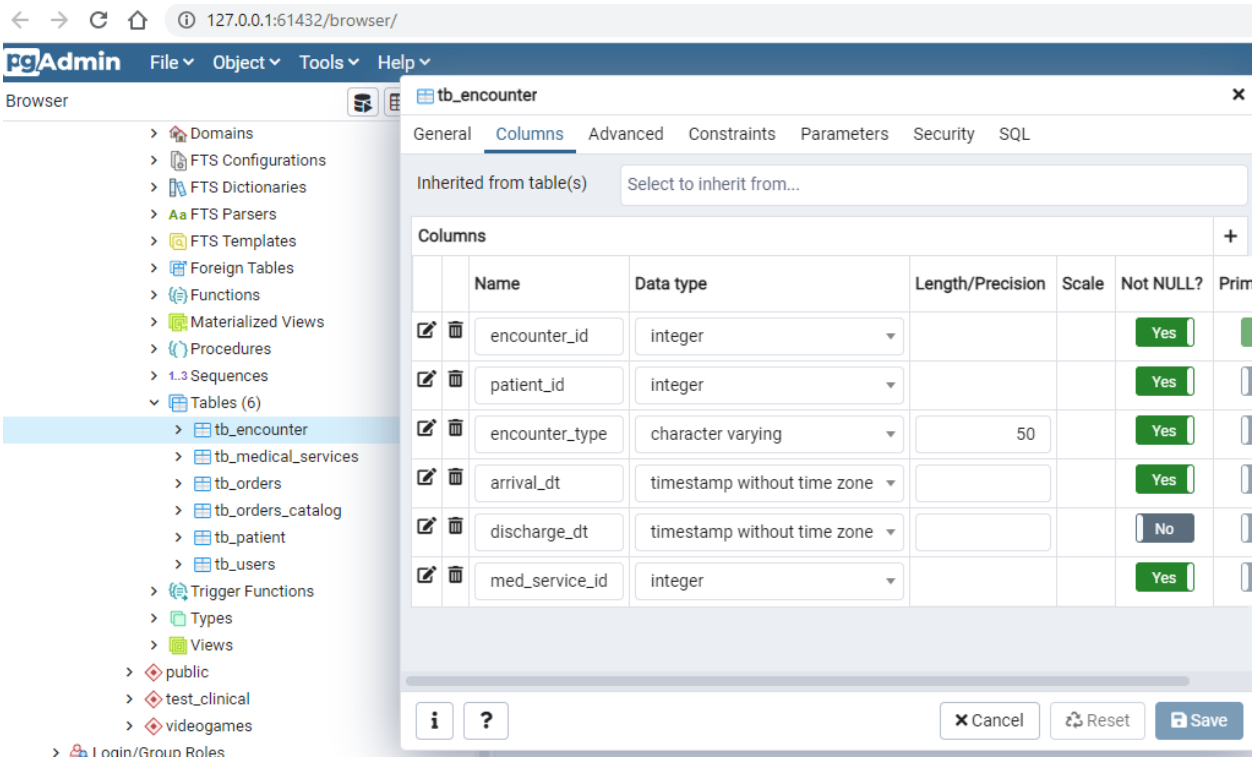


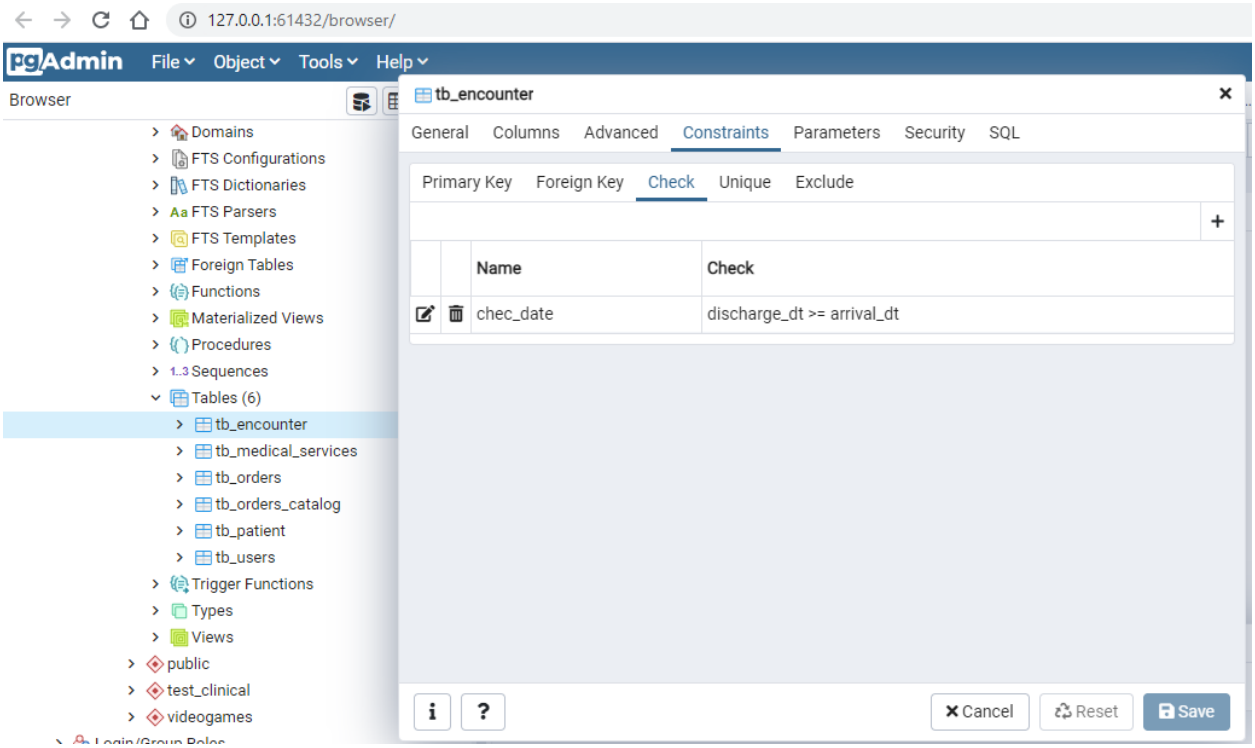
EJERCICIO 3 (30%)

a) Se va a ofrecer un nuevo servicio en la clínica, para ello deberemos añadir una nueva especialidad (*Digestología*) de la cual pasará a encargarse el Dr. Peris, y se ofrecerá una nueva prueba diagnóstica (*Colonoscopia*). Así pues se deben insertar la nueva especialidad y la nueva prueba según la especificación adjunta, y se debe modificar la especialidad del Dr. Peris (con `user_id = 13`) que ahora dirigirá esta nueva especialidad de Digestología. En total deberéis hacer tres sentencias.

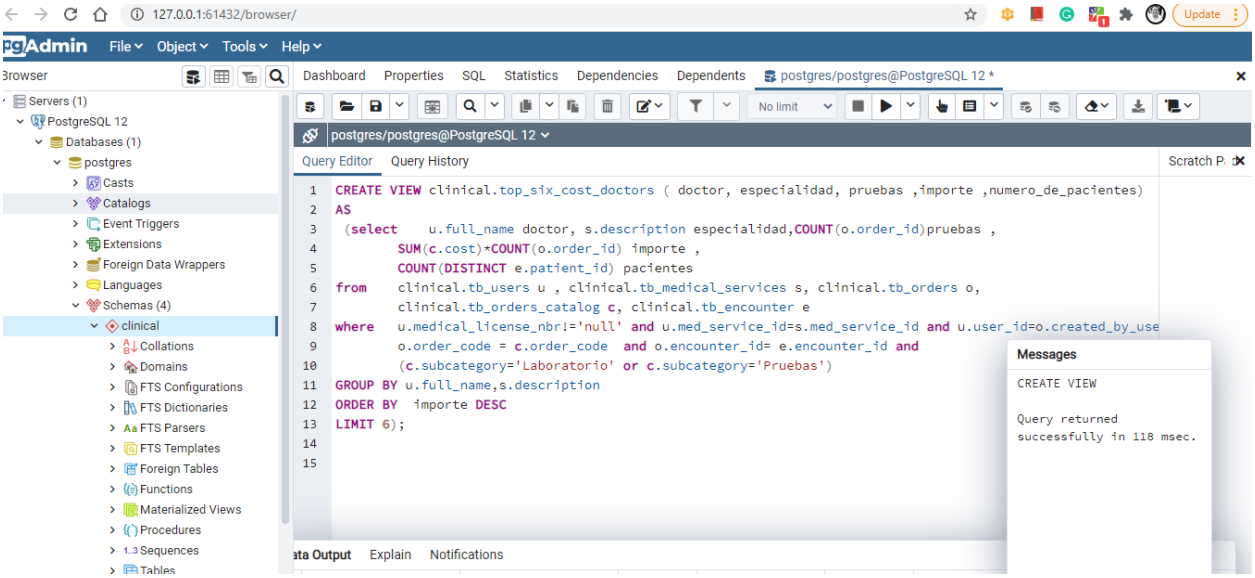


b) Nos comentan que se debe de **alterar**, en una **única sentencia**, la tabla `tb_encounter` para controlar que `discharge_dt` pueda ser nulo, pero nunca sea inferior a `arrival_dt`.



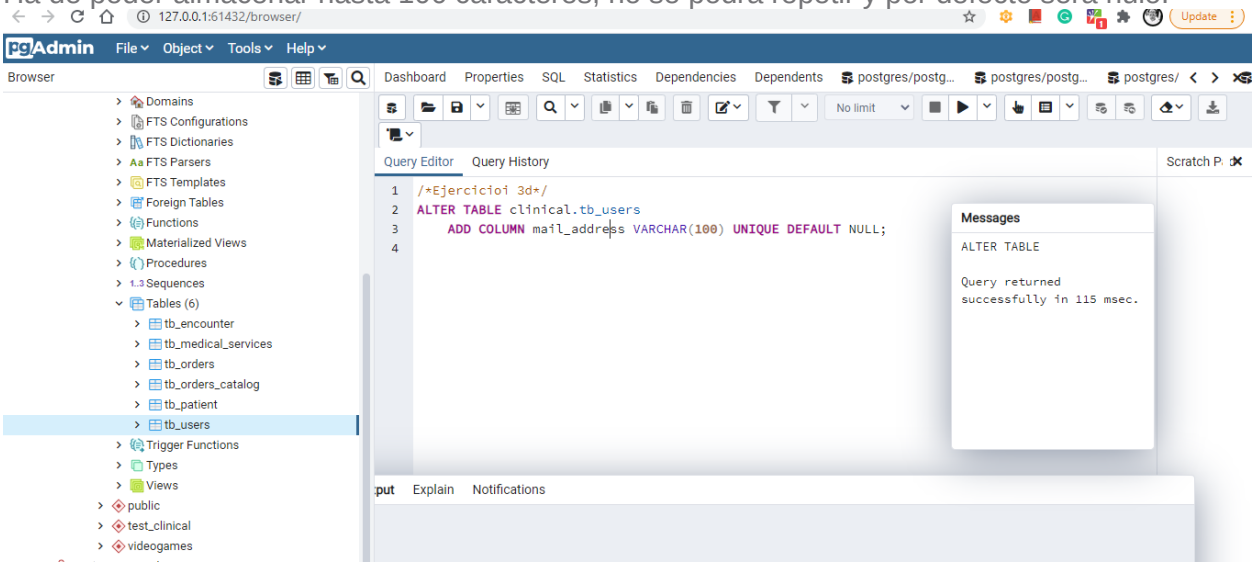


c) Implementad, en una **única sentencia**, una vista **clinical.*top_six_cost_doctors*** sobre la consulta c) del ejercicio 2, pero haciendo que aparezcan 6 en vez de 3 médicos. Aseguraos de que los usuarios no puedan insertar valores que inicialmente no deben mostrarse a través de la vista.



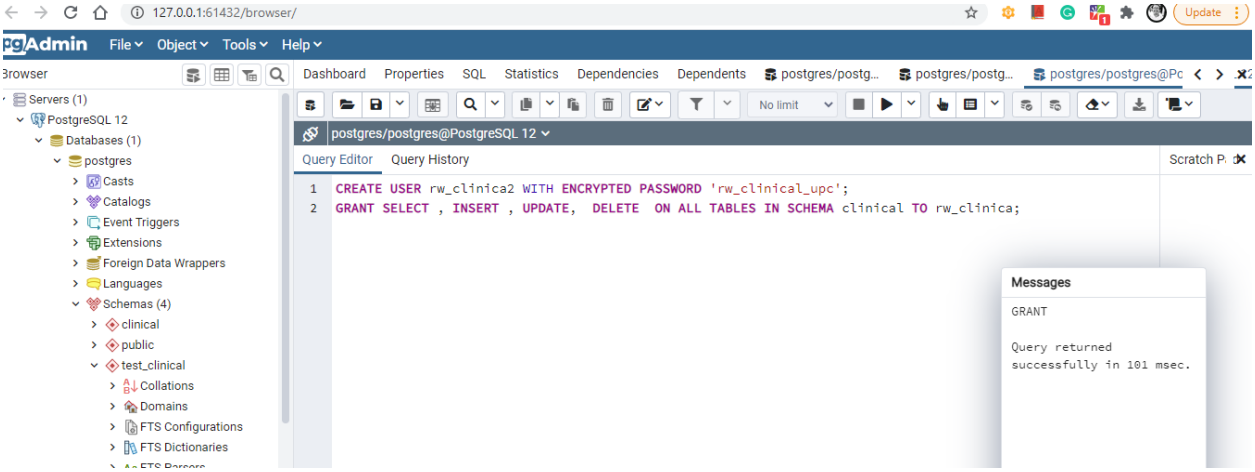
Nota : Esta ultimo requerimiento : “Aseguraos de que los usuarios no puedan insertar valores que inicialmente no deben mostrarse a través de la vista” no entiendo el contexto en el que tiene que ser implementado?

d) Nos piden, que se añada, en una **única sentencia**, una columna en la tabla `tb_users` denominada `mail_address`, para almacenar la dirección de correo electrónico del usuario. Ha de poder almacenar hasta 100 caracteres, no se podrá repetir y por defecto será nulo.



e) Se ha de crear un usuario de sistema (cuidado! no confundir con tb_users) ***rw_clinical*** *con acceso al esquema* ***clinical*** *y* contraseña ***rw_clinical_uoc.*** Este usuario puede realizar lecturas, inserciones, actualizaciones, borrados (nunca truncamientos) de las tablas del esquema ***clinical*** (*rw* significa *read and write*). Asignar los permisos necesarios a dicho usuario mediante SQL. Aseguraos también de que este usuario no pueda asignar permisos a otros usuarios sobre dichas tablas. Referencia a la documentación:

https://www.postgresql.org/docs/12/sql-grant.html



EJERCICIO 4 (20%)

Desde la empresa nos piden que realicemos unas tareas de comprensión y práctica sobre PostgreSQL, como nuevos expertos que sois.
Contestad brevemente a los siguientes apartados:
Explicar en qué consisten los predicados: `LIKE`, `ALL`, `ANY/SOME` y `EXISTS` y poned ejemplos de uso sobre la base de datos de la clínica.

LIKE: Esta es una expresión que nos permite ver si una columna cumple alguna característica determinada ósea retornara true si el contenido de la columna hace matching con la condiciones o patrón especificado y retorna false en caso no, Utilizamos expresiones regulares para trabajar con condiciones complejas : En términos que nos interesa False/True significa devuelve el registro o no

Ejemplo: Nos devuelve todos lo registros de los empleado de la clínica cuyo nombre completo empiece con “Dra” (Nos interesa obtener la información de todas las doctoras)

- SELECT * FROM clinical.tb_users u WHERE u.full_name LIKE 'Dra%'

ALL: Esta es una expresión que nos permite ver si una columna cumple una condición sobre una subquery, la característica de esta expresión es que todos los registros de la subquery tiene que cumplir la condición para devolver TRUE , en casos contrario devuelve FALSE. En términos que nos interesa TRUE/FALSE devuelve el registro o no.

Ejemplo: Nos devuelve todos lo registros de los paciente que no han tenido ninguna servicio de Urgencia (Nos interesa ofrecer un servicio personalizado a un segmento especifico de nuestros clientes)

- SELECT * FROM clinical.tb_patient p WHERE p.patient_id <> ALL (select e.patient_id from clinical.tb_encounter e where e.encounter_type = 'Urgencia' and e.patient_id=p.patient_id)

ANY/SOME: Esta es una expresión que nos permite ver si una columna cumple una condición sobre una subquery, la característica de esta expresión es alguno de los registros de la subquery tiene que cumplir la condición para devolver TRUE , en casos contrario devuelve FALSE. En términos que nos interesa TRUE/FALSE devuelve el registro o no.

Ejemplo: Nos devuelve todos lo registros de los paciente que han algún servicio de Urgencia (Nos interesa ofrecer un servicio personalizado a un segmento especifico de nuestros clientes)

- SELECT * FROM clinical.tb_patient p WHERE p.patient_id = ANY (select e.patient_id from clinical.tb_encounter e where e.encounter_type = 'Urgencia' and
  e.patient_id=p.patient_id)

EXIST: Es una expresión que nos permite ver si existe algún valor en una subquery , la característica de esta expresión es si alguno de los registros de la subquery existe devuelve TRUE en casos contrario devuelve FALSE.

Ejemplo: Nos devuelve todos lo registros de los paciente que han tenido algún servicio de Urgencia (Nos interesa ofrecer un servicio personalizado a un segmento especifico de nuestros clientes)

- SELECT * FROM clinical.tb_patient p WHERE EXISTS (select e.patient_id from clinical.tb_encounter e where e.encounter_type = 'Urgencia' and
  e.patient_id=p.patient_id)

NOTA : Solo comentar que desde mi punto de vista ALL, ANY/SOME / EXIST son bastantes similares, comparan un valor con una lista de valores , podría ser solo aplique sobre una relaciones 1 a N ((diagrama de entidad relación)? … Estas notas las pongo con la intención de recibir feedback, ya que he considero importante el feedback de un profesor experto en mi camino de aprendizaje…Gracias!

Criterios de valoración

En el enunciado se indica el peso/valoración de cada ejercicio.

Para conseguir la puntuación máxima en los ejercicios, es necesario explicar con claridad la solución que se propone.

Formato y fecha de entrega

El formato del archivo que contiene vuestra solución puede ser **.pdf, .doc y .docx**. **Para otras opciones, por favor, contactar previamente con vuestro consultor**. El nombre del fichero debe contener el código de la asignatura, vuestro apellido y vuestro nombre, así como el número de actividad (PEC2).

El fichero .zip que contenga todos los ficheros de la PEC (tanto los ficheros .sql como el documento que muestra los resultados de vuestras soluciones) tenéis que enviarlo al buzón de Entrega y registro de EC disponible en el aula (apartado Evaluación).

La fecha límite para entregar la PEC2 es el **11/04/2021**

Nota: **Propiedad intelectual**

Al presentar una práctica o PEC que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL etc.). El estudiante tendrá que asegurarse que la licencia que sea no impide específicamente su uso en el marco de la práctica o PEC. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por el copyright.

Será necesario, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente, si así corresponde.