



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

NOMBRE Y APELLIDOS: _____

La empresa “UOC Salud”, quiere disponer de una base de datos para registrar toda la información sobre la actividad que realiza. Por este motivo, el departamento de sistemas de información de la empresa se ha puesto en contacto con la UOC y nos ha pedido que sean los alumnos de la asignatura **Bases de datos para *Data Warehousing*** los que se encarguen de este trabajo.

Consideraciones para la entrega y realización de la PEC:

Todo lo que se pide en esta PEC está explicado en el bloque didáctico 2 (salvo que se trate de un ejercicio de investigación, cuyo enunciado lo especificará). No es necesario adelantar el estudio del material de otros bloques didácticos para la realización de esta PEC.

Se recomienda la utilización de **pgAdmin** para la implementación de toda la PEC. Existe otra alternativa que es **psql** (línea de comandos), pero es preferible que utilicéis pgAdmin ya que es una interfaz gráfica que os permitirá editar y crear sentencias SQL (así como mostrar los resultados) de forma más sencilla que **psql**.

Tal y como se indica en el enunciado, cada respuesta a los ejercicios ha de entregarse en un fichero **.sql** diferente, con el nombre correspondiente. Se evaluará el código entregado en estos ficheros **.sql** y **NO el código que aparezca en el documento o en los pantallazos adjuntos**.

Las capturas de pantalla de los ejercicios (y explicaciones pertinentes) han de proporcionarse en un documento aparte (se proporciona una plantilla para el caso, **indicad vuestro nombre en el documento**, por favor).

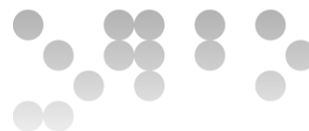
Se debe de realizar la entrega de todos los ficheros de la PEC (tanto los ficheros **.sql** como el documento con explicaciones y capturas de pantalla) en un fichero comprimido **.zip**.

Consideraciones para la evaluación del ejercicio:

Los **scripts** proporcionados por el estudiante con las soluciones de los ejercicios han de ejecutarse correctamente. El estudiante ha de asegurarse de que lanzando el **script** completo de cada ejercicio no produzca ningún error.

Importante: Las sentencias SQL proporcionadas en los **scripts** han de ser creadas de forma manual y no mediante asistentes que PostgreSQL/pgAdmin puedan proporcionar. Se pretende aprender SQL y no la utilización de asistentes.

Las sentencias SQL proporcionadas en los ejercicios han de ser **solamente** aquellas que pide el enunciado y ninguna otra más. Cualquier sentencia añadida a mayores, si está mal o provoca que el **script** no se ejecute correctamente a la hora de corregirlo, penalizará el ejercicio.



EJERCICIO 1 (20%)

Se pide proporcionar las sentencias SQL para crear los componentes especificados a continuación. El resultado de este ejercicio ha de entregarse en el fichero **pec2_ej1.sql**.

BD_DW2

Nombre de la base de datos donde se deben de crear el esquema y los componentes requeridos.

Nota: el comando de creación de la base de datos ha de proporcionarse comentado en el script.

clinical

Nombre del esquema de base de datos donde se deben de crear los componentes especificados a continuación.

tb_medical_services

Tabla para guardar información acerca de las especialidades de que dispone la nueva clínica "UOC Salud". Se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_medical_services Esquema: clinical					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
med_service_id	Númérico entero	No	Si		
description	Cadena de 50 caracteres variable	No	No		
surgical	Carácter	No	No		
short_code	Cadena de 3 caracteres	No	No		

tb_users

Tabla para guardar información sobre los usuarios que interactúan con el sistema. De estos usuarios se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_users Esquema: clinical					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
user_id	Númérico entero	No	Si		
user_name	Cadena de 10 caracteres	No	No		
user_type	Cadena de 10 caracteres	No	No		
full_name	Cadena de 50 caracteres variable	No	No		
register_dt	Fecha con tiempo	No	No		Fecha Actual
medical_license_nbr	Cadena de 10 caracteres	Sí	No		
med_service_id	Númérico entero	Sí	No	tb_medical_services	

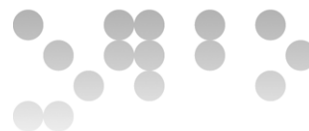
**tb_patient**

Tabla para guardar información acerca de los pacientes de la clínica. Se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_patient Esquema: clinica					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
patient_id	Número entero	No	Si		
ehr_number	Número entero	Si	No		
name	Cadena de 50 caracteres variable	No	No		
sex	Cadena de 1 carácter	Si	No		
birth_dt	Fecha	Si	No		
residence	Cadena de 100 caracteres variable	Si	No		
insurance	Cadena de 50 caracteres variable	Si	No		

tb_encounter

Tabla para guardar información acerca de las atenciones que reciba un paciente en la clínica (episodios). De estos episodios se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_encounter Esquema: clinica					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
encounter_id	Número entero	No	Si		
patient_id	Número entero	No	No	tb_patient	
encounter_type	Cadena de 50 caracteres variable	No	No		
arrival_dt	Fecha con tiempo	No	No		Fecha Actual
discharge_dt	Fecha con tiempo	Si	No		
med_service_id	Número entero	No	No	tb_medical_services	

tb_orders_catalog

Tabla para guardar información acerca del catálogo de las prestaciones que ofrece la clínica. De estas prestaciones se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_orders_catalog Esquema: clinica					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
order_code	Número entero	No	Si		
category	Cadena de 50 caracteres variable	No	No		
subcategory	Cadena de 50 caracteres variable	No	No		
order_desc	Cadena de 50 caracteres variable	No	No		
cost	Número real	No	No		

tb_orders

Tabla para guardar información acerca de las prestaciones que se realizan a los pacientes. De estas prestaciones a pacientes se quiere guardar información según la especificación proporcionada a continuación.

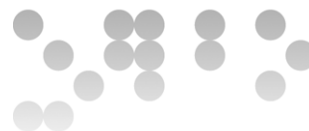


Tabla: tb_orders Esquema: clinical					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
order_id	N Numérico entero	No	Si		
order_code	N Numérico entero	No	No	tb_orders_catalog	
encounter_id	N Numérico entero	No	No	tb_encounter	
status	C Cadena de 50 caracteres variable	No	No		
created_dt	F Fecha con tiempo	No	No		Fecha Actual
status_dt	F Fecha con tiempo	Sí	No		Fecha Actual
created_by_user	N Numérico entero	No	No	tb_users	

Una vez ejecutadas las sentencias SQL de creación de las estructuras que se requieren, se pide ejecutar el script **BBDD_Clinical_data.sql** adjunto. Este fichero NO puede ser modificado para la realización de la PEC y es necesario que todas las sentencias SQL del fichero se ejecuten correctamente. **Si se encuentran indicios de que el fichero se ha modificado o que los datos están incompletos, toda la PEC será considerada como no válida.**

EJERCICIO 2 (30%)

Se pide proporcionar, para cada apartado, **una única consulta en lenguaje SQL**, y el **resultado de dicha consulta mediante una captura del resultado en pgAdmin**. Para poder realizar este ejercicio, el primer ejercicio ha de estar finalizado de forma correcta.

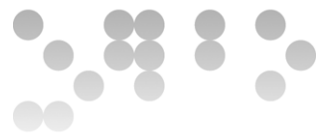
Las consultas SQL de este ejercicio han de entregarse en un fichero llamado **pec2_ej2.sql** (las capturas de pantalla, y explicaciones si son necesarias, han de entregarse en un documento adjunto).

Los resultados de las consultas no deben contener repeticiones en ningún caso, por lo que la eliminación de resultados puede ser necesaria en alguna de las consultas. Por otro lado, el uso innecesario de la cláusula DISTINCT será penalizado.

Nota: La instalación por defecto de PostgreSQL es *case sensitive*, esto es, la cadena de caracteres "ALUMNO UOC" es diferente de "Alumno UOC" y de "alumno uoc".

Nota: Es posible que para alguna de las consultas exista más de una solución. En este caso, cualquiera de las soluciones se considerará válida, siempre y cuando cumpla con las aclaraciones proporcionadas en el enunciado.

- 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.



Messages Data Output

	order_id [PK] integer	encounter_id integer	status character varying (50)	created_dt timestamp without time zone
1	440	349048	Realizada	2019-01-02 10:53:00
2	441	349048	Realizada	2019-01-02 10:53:00
3	442	349048	Realizada	2019-01-02 10:53:00
4	175	920935	Realizada	2019-05-27 12:55:00
5	176	920935	Realizada	2019-05-27 12:55:00
6	177	920935	Realizada	2019-05-27 12:55:00

- 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.

Messages Data Output

	name character varying (50)	category character varying (50)	order_desc character varying (50)	order_id integer	encounter_type character varying (50)	status character varying (50)	status_dt timestamp without time zone
1	Claudia	M.Interna	Consulta sucesiva MIN	357	Urgencia	Cancelada	2009-02-09 14:59:00
2	Ada	Laboratorio	Hemograma..San	361	Consulta Externa	Cancelada	2009-11-13 11:59:00
3	Ada	Laboratorio	Creatinina..Srm	363	Consulta Externa	Cancelada	2009-11-13 11:59:00

- 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.

Data Output Explain Messages Notifications

	full_name character varying (50)	description character varying (50)	orders bigint	patients bigint	cost numeric (5,2)
1	Dra. Barceló	Cirugía General y Digestiva	53	15	350.20
2	Dra. Adriana	Cirugía General y Digestiva	55	15	230.75
3	Dra. Rivera	Radiodiagnóstico	66	18	139.53

- 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.

Messages Data Output

	patient_id [PK] integer	ehr_number integer	name character varying (50)	sex character (1)	birth_dt date	residence character varying (100)	insurance character varying (50)
1	8	1007	Sara	F	1965-11-07	Carrer de dalt	Mapfre
2	31	1030	Josep	M	1949-11-07	Carrer Principal	Mapfre



- 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.

Messages Data Output

	ehr_number integer	name character varying (50)	age double precision	visitas bigint
1	1003	Eva	51	9
2	1014	Antonio	20	10
3	1005	Marta	22	11
4	1015	Ada	10	46

EJERCICIO 3 (30%)

Se pide proporcionar, para cada apartado, **las sentencias en lenguaje SQL, y sus resultados mediante una captura en pgAdmin**. Para poder realizar este ejercicio, el ejercicio 1 ha de estar finalizado.

Las sentencias SQL de este ejercicio han de entregarse en un fichero llamado **pec2_ej3.sql** (las capturas de pantalla han de entregarse en un documento adjunto).

- 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.

Tabla: tb_medical_services Esquema: clinical		Tabla: tb_orders_catalog Esquema: clinical	
Nombre columna	Datos	Nombre columna	Datos
med_service_id	6	order_code	9001
description	Digestología	category	M.Interna
surgical	S	subcategory	Endoscopia
short_code	DIG	order_desc	Colonoscopia
		cost	600,00

- 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.



- 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>

Nota: podéis probar los permisos creando una nueva conexión con el usuario creado y mantener así las dos sesiones abiertas. Para ello, debéis crear un nuevo servidor en el Browser de pgAdmin con las credenciales del nuevo usuario.

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:

- 1) 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.

a) El predicado `LIKE` puede ser muy útil si intentamos hacer búsquedas en columnas de tipo cadena de carácter, y no conocemos el valor exacto de la cadena:

Por ejemplo, si queremos conocer de las pruebas posibles (`tb_orders_catalog`) aquellas en las que figura RX en la descripción (`order_desc`, `cost`), podemos ejecutar:

```
SELECT * FROM tb_orders_catalog WHERE order_desc LIKE '%RX%';
```

Messages		Data Output			
	order_code [PK] integer	category character varying (50)	subcategory character varying (50)	order_desc character varying (50)	cost real
1	2292	Radiología	Radiología Convencional	RX Tobillo AP/L Dcho	38.3
2	2293	Radiología	Radiología Convencional	RX Tobillo AP/L Izdo	38.3
3	2294	Radiología	Radiología Convencional	RX Tobillo AP/L Bilateral	38.3
4	9000	Radiología	Radiología Convencional	RX Tórax PA/L	24.7
5	1001	Radiología	Radiología Convencional	RX Tórax AP Decúbito	24.7

- b) El predicado `ALL` se utiliza para ver si una columna cumple que todas sus filas satisfagan una condición.



Si queremos conocer aquellas pruebas que tienen un coste superior a todas las pruebas de la categoría Radiología, efectuaremos la consulta:

```
SELECT *
FROM tb_orders_catalog
WHERE cost > ALL(SELECT cost
                  FROM tb_orders_catalog
                  WHERE category = 'Radiología');
```

Messages Data Output

	order_code [PK] integer	category character varying (50)	subcategory character varying (50)	order_desc character varying (50)	cost real
1	9001	M.Interna	Endoscopia	Colonoscopia	600

c) El predicado ANY/SOME se utiliza para ver si una columna cumple que alguna o algunas de sus filas satisfagan una condición.

Si queremos conocer aquellas pruebas que tienen un coste superior a alguna de las pruebas de la subcategoría Cardioversión, efectuaremos la consulta:

```
SELECT *
FROM tb_orders_catalog
WHERE cost > ANY (SELECT cost
                  FROM tb_orders_catalog
                  WHERE subcategory = 'Cardioversión');
```

Messages Data Output

	order_code [PK] integer	category character varying (50)	subcategory character varying (50)	order_desc character varying (50)	cost real
1	1368	Cardiologia	Esfuerzo	Prueba de esfuerzo	168.18
2	1750	Laboratorio	Laboratorio	Biopsia Páncreas ,Estudio A...	163.22
3	6711	M.Interna	Ecografía	MIN Ecografía intervencioni...	214.87
4	6712	M.Interna	Ecografía	MIN Ecografía doppler de tr...	136.95
5	3975	M.Interna	Ecografía	Ecografía venosa de compre...	136.95
6	6710	M.Interna	Ecografía	MIN Ecografía venosa de co...	136.95
7	1365	M.Interna	Pruebas	Paracentesis	141.49
8	1366	M.Interna	Pruebas	Toracocentesis	209.93

d) EXISTS (o test de existencia) se utiliza para ver si una subconsulta produce alguna fila de resultados, normalmente relacionada mediante alguna columna con la consulta. Si necesitamos comprobar que una subconsulta no produce ninguna fila de resultados, podemos utilizar NOT EXISTS.

Si queremos saber los médicos que no han pedido nunca una prueba de Radiología, podríamos hacer:

```
SELECT * FROM tb_users u
WHERE user_type = 'Asist' AND NOT EXISTS (SELECT *
```




```
FROM tb_orders o NATURAL JOIN tb_catalog_orders ca
WHERE o.created_by_user = u.user_id AND ca.category =
'Radiología');
```

Messages Data Output

	user_id [PK] Integer	user_name character (10)	user_type character (10)	full_name character varying (50)	register_dt timestamp without time zone	medical_license_nbr character (10)
1	1	11133355A	Asist	Dr. Martí	2015-03-01 00:00:00	46111A
2	2	22244466B	Asist	Dr. López	2017-01-01 00:00:00	46222M
3	3	12312344C	Asist	Dra. Pintos	2018-08-01 00:00:00	30220C
4	4	33345665D	Asist	Dra. Adriana	2015-03-01 00:00:00	43881R
5	5	77791982E	Asist	Dra. Barceló	2017-01-01 00:00:00	48932A
6	6	44332772F	Asist	Dra. Rivera	2020-02-01 00:00:00	46555T
7	7	88544588G	Asist	Dra. Palop	2016-01-01 00:00:00	47830B
8	8	99853585H	Asist	Dr. Abad	2015-07-01 00:00:00	36190A

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.

