

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

NOMBRE Y APELLIDOS: _____

El Comité Olímpico UOC, quiere disponer de una base de datos para registrar toda la información sobre los Juegos Atléticos Olímpicos UOC. 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.

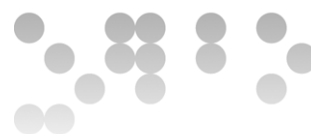
Consideraciones para la entrega y realización de la PEC:

- 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 utilizéis 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 (25%)

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**.

pec2

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.

olimpic

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

tb_discipline

Tabla para guardar información acerca de las distintas disciplinas deportivas de los juegos olímpicos. Se quiere guardar información según la especificación proporcionada a continuación. El tipo de disciplina (*type*) solamente puede tener los valores RUN, JUMP o THROW.

Tabla: tb_discipline Esquema: olimpic					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
discipline_id	Númérico entero	No	Si		
name	Cadena de 50 caracteres variable	No	No		
type	Cadena de 10 caracteres variable	No	No		
object_type	Cadena de 20 caracteres fija	No	No		

tb_athlete

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

Tabla: tb_athlete Esquema: olimpic					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
athlete_id	Cadena de 7 caracteres fija	No	Si		
name	Cadena de 50 caracteres variable	No	No		
country	Cadena de 3 caracteres fija	No	No		
substitute_id	Cadena de 7 caracteres fija	Si	No	tb_athlete	

tb_play

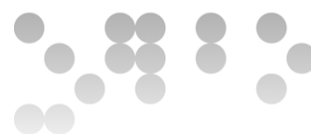


Tabla para guardar información acerca de las disciplinas de los atletas. Se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_play Esquema: olimpic					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
athlete_id	Cadena de 7 caracteres fija	No	Si	tb_athlete	
discipline_id	Numérico Entero	No	Si	tb_discipline	

tb_round

Tabla para guardar información acerca de las tandas. Se quiere guardar información según la especificación proporcionada a continuación.

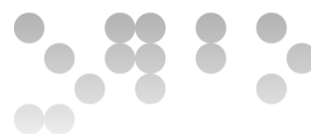
Tabla: tb_round Esquema: olimpic					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
round_number	Numérico Entero	No	Si		
discipline_id	Numérico Entero	No	Si	tb_discipline	

tb_register

Tabla para guardar información acerca de resultados de los atletas en una tanda. Se quiere guardar información según la especificación proporcionada a continuación.

Tabla: tb_register Esquema: olimpic					
Nombre columna	Tipo de datos	Acepta Nulos	Clave primaria	Clave foránea	Valor por defecto
athlete_id	Cadena de 5 caracteres fija	No	Si	tb_athlete	
round_number	Numérico entero	No	Si	tb_round	
discipline_id	Numérico entero	No	Si	tb_round	
register_date	Fecha	No	No		Fecha Actual
register_position	Numérico entero	Si	No		
register_time	Tiempo	Si	No		
register_measure	Real	Si	No		

Una vez ejecutadas las sentencias SQL de creación de las estructuras que se requieren, se pide ejecutar el script **BBDD_Olimpic_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 (25%)

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).

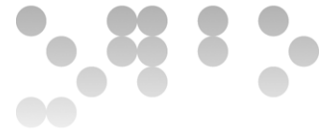
Los resultados de las consultas no deben contener repeticiones en ningún caso, por lo que la eliminación de resultados puede ser necesaria en alguna de las consultas. Por otro lado, el uso innecesario de la cláusula DISTINCT será penalizado.

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 los atletas españoles cuyo nombre empieza por PE, ordenados descendientemente por su identificador. Queremos obtener el identificador, nombre, país e identificador del sustituto.

	athlete_id [PK] character (7)	name character varying (50)	country character varying (3)	substitute_id character (7)
1	1358251	PENA Inigo	ESP	1302771
2	1349249	PEREZ Maria Isabel	ESP	1307950
3	1349246	PEREZ Marta	ESP	1324396
4	1349245	PEREZ Maria	ESP	1334678
5	1349243	PELETEIRO Ana	ESP	1304891
6	1325882	PEREZ BLANCO Jimena	ESP	1302772



- b) Proporcionar la lista de atletas franceses que practican alguna disciplina de salto, ordenados ascendentemente por el nombre de la disciplina y descendientemente por su nombre. Queremos obtener el nombre de disciplina, nombre del atleta e identificador del atleta.

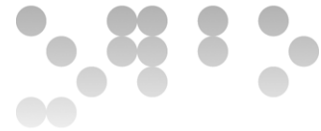
	name character varying (50) 🔒	name character varying (50) 🔒	athlete_id character (7) 🔒
1	BMX Cycling	VALENTINO Manon	1319979
2	BMX Cycling	MAHIEU Romain	1319977
3	BMX Cycling	ETIENNE Axelle	1319978
4	BMX Cycling	DAUDET Joris	1319973
5	BMX Cycling	ANDRE Sylvain	1319971
6	Trampoline	MORANTE Allan	1334953
7	Trampoline	LABROUSSE Lea	1334991

- c) Proporcionar la disciplina con más atletas participantes. Queremos obtener el identificador de la disciplina, el nombre y el total de atletas.

	discipline_id [PK] integer	name character varying (50)	count bigint 🔒
1	2	Athletics	2041

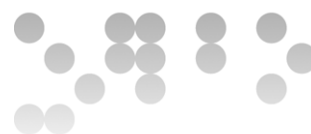
- d) Proporcionar la lista de atletas que participan en más de una disciplina. Queremos ver el id de los atletas, el nombre, el país y el total de disciplinas que practica.

	athlete_id [PK] character (7)	name character varying (50)	country character varying (3)	count bigint 🔒
1	1467919	HAVIK Yoeri	NED	2
2	1330503	BRENNAUER Lisa	GER	2
3	1346205	GANNA Filippo	ITA	2
4	1330510	KLEIN Lisa	GER	2
5	1342530	DYGERT Chloe	USA	2
6	1393561	KOPECKY Lotte	BEL	2



- e) Proporcionar la lista de atletas que en más tandas han participado. Queremos ver el id de los atletas, el nombre, la disciplina y el total de tandas en las que han participado.

	athlete_id character (7)	name character varying (50)	discipline_id integer	name character varying (50)	count bigint
1	1478152	PROSTOROV Mykola	15	Trampoline	4
2	1429192	KOCHESOK Susana	15	Trampoline	4



EJERCICIO 3 (25%)

Se pide proporcionar, para cada apartado, **una única sentencia en lenguaje SQL siempre que sea posible**, y el resultado de dicha sentencia mediante una captura del resultado 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).

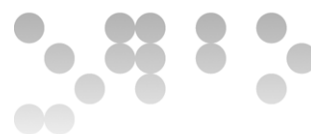
- a) Nos dicen que hace falta insertar 3 nuevos atletas según la especificación dada.

Nombre columna	Atleta 1	Atleta 2	Atleta 3
athlete_id	0000001	0000002	0000003
name	REMBRAND Luc	SMITH Mike	LEWIS Carl
country	FRA	ENG	USA
substitute_id	NULL	NULL	NULL

- b) Nos piden que añadamos una nueva restricción para impedir que haya atletas españoles sin sustituto.
- c) Implementad una vista **olimpic.exercise33** sobre la consulta a) del ejercicio 2. 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 una columna en la tabla de tb_athlete denominada `date_add`, que representa la fecha de alta en la federación de su país. Este campo no puede tener valores nulos y, por defecto, tomará la fecha actual.
- e) Se ha de crear un usuario de sistema (cuidado! no confundir con tb_user) **registerer con acceso al esquema olimpico** y contraseña **1234**. Este usuario puede realizar lecturas, inserciones, actualizaciones, borrados (nunca truncamientos) de la tabla tb_register, y permisos de lectura en la tabla tb_athlete. No podrá acceder a las otras tablas del esquema **olimpic**. 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 (25%)

Desde la empresa nos piden que realicemos unas tareas de investigación sobre PostgreSQL, ya que no disponen de un experto en estos momentos para poder realizarlas.

Contestad brevemente a los siguientes apartados:

- a) Explicar las cláusulas GROUPING SET, ROLLUP y CUBE de PostgreSQL. La explicación debe ser complementada además con ejemplos de elaboración propia. **(15%)**

Estas cláusulas permiten definir distintos tipos de agrupamiento (GROUP BY).

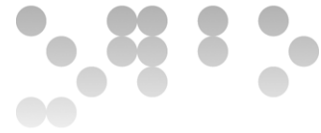
- **GROUPING SET:** equivalente a hacer distintos agrupamientos simples GROUP BY y combinar los resultados. El resultado contiene todos los campos de los distintos agrupamientos con valores NULL cuando no son utilizados.

```
SELECT country, substitute_id, COUNT(*)
FROM olimpib.tb_athlete
WHERE country IN ('ESP','FRA','ENG') AND substitute_id IS NOT NULL
GROUP BY GROUPING SETS((country),(substitute_id))
HAVING substitute_id IS NULL OR count(*)>2
```

	country character varying (3)	substitute_id character (7)	count bigint
1	ESP	[null]	132
2	ENG	[null]	1
3	FRA	[null]	188
4	[null]	1889067	4
5	[null]	1341483	3
6	[null]	0000001	4
7	[null]	1307950	3

- **ROLLUP:** dada una lista de campos genera el GROUPING SET de todos los subconjuntos de campos continuos empezando por la izquierda. Por ejemplo:

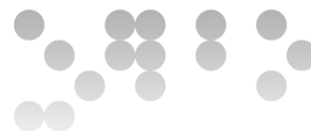
```
ROLLUP ( e1, e2, e3, ... )
```

```
GROUPING SETS (
    ( e1, e2, e3, ... ),
    ...
    ( e1, e2 ),
    ( e1 ),
    ( )
)
```

```
SELECT country, substitute_id, COUNT(*)
FROM olimpico.tb_athlete
WHERE country IN ('ESP','FRA','ENG') AND substitute_id IS NOT NULL
GROUP BY ROLLUP (country,substitute_id)
HAVING substitute_id IS NULL OR count(*)>1
```

	country character varying (3)	substitute_id character (7)	count bigint
1	[null]	[null]	321
2	ESP	1485432	2
3	FRA	1324460	2
4	ESP	1476208	2
5	ESP	1341483	2
6	ESP	0000001	2
7	FRA	1360801	2
8	FRA	1337490	2
9	FRA	1347044	2
10	FRA	1307950	2
11	ESP	1285817	2
12	FRA	1393269	2
13	ESP	1889067	2
14	FRA	1333346	2
15	FRA	1330891	2
16	FRA	1431591	2
17	FRA	1485427	2
18	FRA	1429019	2
19	ESP	1333472	2
20	FRA	1889067	2
21	ESP	[null]	132
22	ENG	[null]	1
23	FRA	[null]	188



- CUBE: la cláusula CUBE permite generar todas las combinaciones de los campos indicados.

```
CUBE ( a, b, c )
```

```
GROUPING SETS (  
  ( a, b, c ),  
  ( a, b   ),  
  ( a,    c ),  
  ( a      ),  
  (    b, c ),  
  (    b   ),  
  (      c ),  
  (        )  
)
```

Referencias:

<https://www.postgresql.org/docs/12/queries-table-expressions.html#QUERIES-GROUPING-SETS>
<https://www.postgresqltutorial.com/postgresql-grouping-sets/>

- b) Explicar las columnas generadas (cláusula GENERATED ALWAYS AS) de PostgreSQL. La explicación debe ser complementada además con ejemplos de elaboración propia. (10%)

Permite definir columnas calculadas en una tabla. Una columna calculada es aquella cuyo valor se calcula a partir del valor de otras columnas no calculadas y funciones inmutables. No pueden formar parte de la clave primaria pero sí de una clave foránea.

La cláusula GENERATED ALWAYS puede aparecer en las sentencias CREATE TABLE o ALTER TABLE.

La cláusula STORED es obligatoria, e indica que los valores de la columna generada se guardan en disco. Otros SGBD como MySQL permiten la opción VIRTUAL, que calcula el valor de la columna calculada en el aire.

```
CREATE TABLE olympic.people (  
  id INT NOT NULL,  
  first_name CHARACTER VARYING(50) NOT NULL,  
  last_name CHARACTER VARYING(50) NOT NULL,  
  full_name CHARACTER VARYING(100) GENERATED ALWAYS AS (first_name||' '||last_name) STORED,  
  CONSTRAINT pk_people PRIMARY KEY (id)  
);
```



```
INSERT INTO olympic.people VALUES (1, 'John', 'Smith');
```

```
SELECT * FROM olympic.people;
```

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	full_name character varying (100)
1	1	John	Smith	John Smith

```
ALTER table olympic.tb_athlete
```

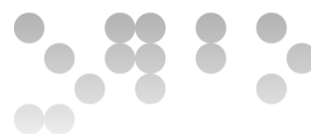
```
ADD COLUMN description VARCHAR(100) GENERATED ALWAYS AS (country || '-' || name) STORED;
```

	athlete_id [PK] character (7)	name character varying (50)	country character varying (3)	substitute_id character (7)	date_add date	description character varying (100)
1	1346266	AALERUD Katrine	NOR	[null]	2021-10-04	NOR-AALERUD Katrine
2	1344792	ABASS Abobakr	SUD	[null]	2021-10-04	SUD-ABASS Abobakr
3	1328854	ABDALLA Abubaker Haydar	QAT	[null]	2021-10-04	QAT-ABDALLA Abubaker Haydar
4	1306332	ABDEL LATIF Radwa	EGY	[null]	2021-10-04	EGY-ABDEL LATIF Radwa
5	1306339	ABDEL RAZEK Samy	EGY	1306332	2021-10-04	EGY-ABDEL RAZEK Samy
6	1312038	ABDELRAHMAN Ihab	EGY	1328854	2021-10-04	EGY-ABDELRAHMAN Ihab
7	1305691	ABDELWAHED Ahmed	ITA	1312038	2021-10-04	ITA-ABDELWAHED Ahmed

Referencias:

<https://www.postgresql.org/docs/12/ddl-generated-columns.html>

<https://severalnines.com/database-blog/overview-generated-columns-postgresql>



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

Tenéis que enviar la PEC al buzón de Entrega y registro de EC disponible en el aula (apartado Evaluación). El formato del archivo que contiene vuestra solución puede ser **.pdf, .doc y .docx**.

Para otras opciones, por favor, contactar previamente con vuestro consultor.

La fecha límite para entregar la PEC2 es el **8/11/2021**.

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.