



## ORDINARIO BASES DE DATOS (26-05-2025)

NOMBRE: \_\_\_\_\_ GRUPO: \_\_\_\_\_

### MODELO E/R (RA6 10 PUNTOS)

Se desea desarrollar un sistema de gestión para un club de fútbol base que administre todas las actividades relacionadas con sus equipos y el entorno que la rodea de manera eficiente.

En primer lugar, es importante registrar información sobre los equipos que forman parte del club, como su nombre, la categoría en la que juegan (por ejemplo, alevín, infantil, cadete o juvenil) y el horario de entrenamiento que tienen asignado. Además, se necesita distinguir las categorías a las que pertenecen estos equipos, indicando un rango de edades para cada una. También existe la posibilidad de que haya categorías que no tengan equipos asociados en ciertos momentos del año.

En cuanto a las instalaciones del club, se debe incluir información sobre los campos de fútbol, registrando su nombre, ubicación y capacidad para espectadores, ya que serán utilizados para entrenamientos y para partidos. En caso de entrenamientos, puede que no se entrene siempre en el mismo campo y queremos constancia de la fecha en que campo entrenaron.

Los jugadores son una parte fundamental, y se debe almacenar información relevante como su nombre, apellidos, fecha de nacimiento, la categoría a la que pertenecen y el equipo en el que juegan, así como su posición en el campo (por ejemplo, portero, defensa, mediocampista o delantero). Además, los jugadores están vinculados a sus padres, madres o tutores, de quienes se debe registrar información como nombre, teléfono de contacto y su relación con el jugador. Hay que tener en cuenta que, en ocasiones, un tutor puede estar relacionado con varios jugadores, por ejemplo, en el caso de hermanos que juegan en diferentes equipos o categorías.

El sistema también debe gestionar las cuotas o tarifas que pagan los padres para que sus hijos jueguen en el club. De cada pago realizado se debe registrar la cantidad, la fecha, el estado (pagado o pendiente) y el método utilizado.

En relación con la gestión del club, se debe incluir información sobre los directivos, como sus nombres, apellidos y el cargo que ocupan (por ejemplo, presidente, secretario o tesorero). Aunque los directivos no están directamente involucrados con los equipos y jugadores en el día a día, son responsables de decisiones como la programación de entrenamientos y la organización de eventos, así como la firma de contratos con proveedores.

Los equipos del club participan en una liga organizada por la federación, que gestiona los partidos, los árbitros, la clasificación y la tabla de goleadores.

Además, el sistema deberá contemplar a los proveedores de material deportivo con los que el club tiene acuerdos. De ellos, se guardará información sobre el tipo de material que suministran (como balones, camisetas o botas), así como sus datos de contacto, como además la cantidad, precio y la fecha de cuando lo suministran. Los materiales se reparten a través del directivo que hizo la firma con el proveedor.

Se quiere crear un sistema flexible y preciso que refleje la realidad de un club de fútbol base en su totalidad, permitiendo una gestión eficiente de las dinámicas internas.

**Representa el esquema conceptual con sus relaciones y cardinalidades, también seleccionar los atributos clave de cada entidad**

Usuario: olimpiada\_

Contraseña: \_\_\_\_\_

IP: 10.81.107.200



### DDL ( RA2 10 PUNTOS)

1. Crea una tabla llamada Empleados que cumpla los siguientes requisitos: deberá tener una columna **id\_empleado** de tipo entero, que actuará como clave primaria y se genera si no le pasas valor o el valor por defecto. La columna **dni** debe ser un texto de 9 caracteres, obligatorio, con valor único, y debe cumplir el formato de un DNI español, compuesto por 8 dígitos seguidos de una letra mayúscula (**por ejemplo, 12345678Z**). Esta validación se implementará mediante una restricción **CHECK** que utilice una expresión regular. La **columna** nombre será un texto de hasta 100 caracteres y también será obligatoria. El campo **salario** debe ser un valor decimal con 8 dígitos en la parte entera y 2 decimales, obligatorio y con la condición de que su valor sea superior a 1000. La columna **fecha\_contratación** será de tipo fecha y su valor será obligatorio. El campo **departamento** debe ser un texto de hasta 50 caracteres y tendrá como valor por defecto "Sin asignar". La columna categoría será un texto limitado a aceptar solo uno de los siguientes valores: '**junior**', '**senior**' o '**manager**'. Por último, el campo activo será de tipo booleano y su valor por defecto será **TRUE**. (RA2 2.5 puntos)
2. Diseña un esquema de base de datos que incluya dos tablas: una tabla llamada **Depart**, con un campo **id\_departamento** de tipo entero como clave primaria, y un campo **nombre** de tipo texto que no pueda ser nulo y cuyos valores sean únicos; y una tabla llamada **Emple**, con un campo **id\_empleado** de tipo entero como clave primaria, un campo **nombre** de tipo texto obligatorio, y un campo **id\_departamento** de tipo entero que actúe como clave foránea referida al campo id\_departamento de la tabla Departamentos. Además, establece las siguientes restricciones: si se elimina un departamento, los empleados asociados deben conservarse, pero su campo id\_departamento debe establecerse a NULL; y si se modifica el valor de id\_departamento en la tabla Departamentos, dicho cambio debe propagarse automáticamente a la tabla Empleados. (RA2 2.5 puntos)
3. Crea una tabla llamada **Inscripciones** para gestionar la participación de los alumnos en los cursos de un sistema de formación. Cada alumno podrá inscribirse en varios cursos, pero solo una vez por curso, por lo que la tabla deberá tener una clave primaria compuesta por los campos **id\_alumno** e **id\_curso**. Ambos campos serán de tipo entero y no podrán ser nulos. La columna **fecha\_inscripcion** será de tipo fecha, su valor será obligatorio y tendrá como valor por defecto la fecha actual. El campo **estado** será de tipo texto y su valor deberá estar restringido a uno de los siguientes: '**pendiente**', '**aceptada**' o '**rechazada**', mediante una restricción **CHECK**. Además, se deben establecer relaciones de integridad referencial con las tablas Alumnos(id\_alumno) y Cursos(id\_curso), de forma que, si se elimina un alumno o un curso, se eliminen automáticamente todas sus inscripciones asociadas. Las tablas Alumnos y Cursos tienen la siguiente estructura: (RA2 2.5 puntos)(El alumnos las crea)

```
CREATE TABLE Alumnos (  
    id_alumno INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE Cursos (  
    id_curso INT PRIMARY KEY,  
    titulo VARCHAR(100) NOT NULL  
);
```

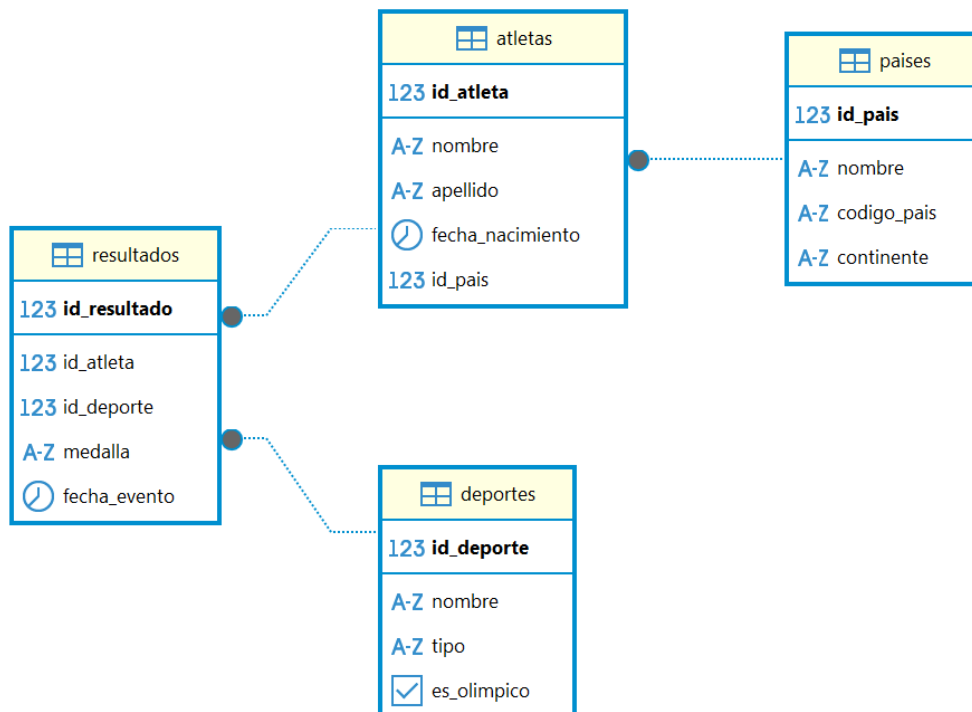
4. Partiendo de la siguiente tabla, la cual debe crear el alumno:



```
CREATE TABLE misalumnos (
  id_alumno INT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL
);
```

Realiza diversas modificaciones utilizando sentencias DDL. En primer lugar, debes añadir un campo llamado **fecha\_nacimiento** de tipo DATE, que sea obligatorio. A continuación, incorpora un nuevo campo **edad**, de tipo entero, y añade una restricción que solo permita introducir valores comprendidos entre 18 y 100 años, ambos inclusive. Posteriormente, añade el campo **teléfono**, de tipo **VARCHAR(9)**, cuya validación debe asegurarse de que empiece por 6 o por 7 y contenga exactamente 9 dígitos; para ello, deberás utilizar una restricción CHECK basada en una expresión regular. Finalmente, crea un índice llamado **ind\_nombre** sobre el campo nombre con el objetivo de optimizar las búsquedas realizadas sobre dicho campo. **(RA2 2.5 puntos)**

### DQL (RA3 8.5 PUNTOS)



5. Queremos obtener un informe que muestre cuántos atletas de cada continente han ganado una medalla de oro en los eventos de las olimpiadas, así como los países del continente que la han obtenido. **(RA3 2 puntos) (string\_agg)**

```
continente|medallas_totales|Países
-----+-----+-----
América  |14 oros      |Brasil, Canadá, Chile, Colombia, Cuba, Estados Unidos, Jamaica, México, Perú|
Asia     |3 oros       |China, India|
Europa   |3 oros       |España|
3 row(s) fetched.
```

6. Queremos un informe que muestre el número total de medallas (oro, plata y bronce) ganadas por continente. Además, incluiremos un ranking. **(RA3 1.5 puntos)**



```

continente|total_medallas|ranking|
-----+-----+-----+
AMÉRICA  |          34|      1|
ASIA     |          14|      2|
ÁFRICA   |           8|      3|
EUROPA   |           4|      4|

```

4 row(s) fetched.

7. Queremos calcular el promedio de edad de los atletas que han ganado medallas (oro, plata o bronce) en el momento del evento. (RA3 1.5 puntos)

```

Tipo medalla |Promedio de Edad|Total de atletas|
-----+-----+-----+
Medalla de Plata |41.7 años      |          20|
Medalla de Oro  |39.7 años      |          20|
Medalla de Bronce|38.8 años      |          20|

```

3 row(s) fetched.

8. Queremos identificar a los atletas que ganaron una medalla el mismo día de su cumpleaños (misma fecha y mes, ignorando el año), mostrando su nombre completo, tipo de medalla y la edad que tenían ese día. (RA3 1.5 puntos).

```

nombre_completo|tipo_medalla|edad_en_evento|
-----+-----+-----+
Mijain López   |oro         |          42|

```

1 row(s) fetched.

9. Queremos listar los países que han ganado medallas, ordenadas por oro, luego por plata y finalmente por bronce (RA3 2 puntos)

```

pais |oros|platas|bronces|
-----+-----+-----+
España | 3| 1| 0|
Cuba   | 2| 4| 0|
China  | 2| 2| 2|
Chile  | 2| 1| 1|
Canadá | 2| 0| 1|
Colombia | 2| 0| 0|
Brasil | 2| 0| 0|
Estados Unidos| 1| 2| 0|
México | 1| 1| 2|
India  | 1| 1| 2|
Jamaica | 1| 0| 3|
Perú   | 1| 0| 0|
Argentina | 0| 3| 2|
Japón  | 0| 3| 1|
Egipto | 0| 1| 1|
Kenia  | 0| 1| 0|
Argelia | 0| 0| 2|
Ghana  | 0| 0| 1|
Sudáfrica | 0| 0| 1|
Marruecos | 0| 0| 1|

```

20 row(s) fetched.

#### DML (RA3 1.5 RA4 10)

10. Queremos actualizar los nombres de los países a mayúsculas, para aquellos países que han ganado medallas en más de 5 deportes diferentes. (RA3 0.75 puntos , RA4 2 puntos)

```

id_pais|nombre |codigo_pais|continente|
-----+-----+-----+
1|Nigeria |NGA        |África    |
2|Kenia    |KEN        |África    |
3|Sudáfrica|ZAF        |África    |

```



4 Egipto	EGY	África	
5 Ghana	GHA	África	
6 Marruecos	MAR	África	
7 Argelia	DZA	África	
8 Estados Unidos	USA	América	
9 Brasil	BRA	América	
10 México	MEX	América	
11 Canadá	CAN	América	
12 Argentina	ARG	América	
13 Colombia	COL	América	
14 Chile	CHL	América	
15 Perú	PER	América	
16 Jamaica	JAM	América	
19 Japón	JPN	Asia	
20 India	IND	Asia	
21 Corea del Sur	KOR	Asia	
22 Rusia	RUS	Asia	
23 Irán	IRN	Asia	
24 Tailandia	THA	Asia	
25 Indonesia	IDN	Asia	
26 Kazajistán	KAZ	Asia	
27 Filipinas	PHL	Asia	
31 Australia	AUS	Australia	
32 Nueva Zelanda	NZL	Australia	
33 Fiyi	FJI	Australia	
34 Alemania	DEU	Europa	
35 Francia	FRA	Europa	
36 Reino Unido	GBR	Europa	
37 Italia	ITA	Europa	
39 España	ESP	Europa	
38 Polonia	POL	Europa	
40 Ucrania	UKR	Europa	
41 Países Bajos	NLD	Europa	
42 Suecia	SWE	Europa	
43 Noruega	NOR	Europa	
17 CUBA	CUB	América	
18 CHINA	CHN	Asia	

40 row(s) fetched.

11. Queremos cambiar el país de los atletas que han ganado un mínimo de dos medallas a un país ficticio **"Olímpicos"** con id\_pais y código de país con valor **999**, y continente **"Olimpia"**, simulando así una nacionalidad honorífica. Todo en un solo comando. **(RA3 0.75 RA4 2 puntos)**

id_atleta	nombre	apellido	fecha_nacimiento	id_pais
152	Shelly-Ann	Fraser-Pryce	1986-12-27	999
61	Taoufik	Makhloufi	1988-04-29	999
171	Liu	Xiang	1983-07-13	999

3 row(s) fetched.

12. Queremos cambiar las medallas de plata a oro para deportistas que tenían menos de 30 años en el momento de ganarlas, como un incentivo a la juventud. **(RA4 2 puntos)**

nombre	apellido	fecha_nacimiento	medalla	fecha_evento
Farida	Osman	1995-01-18	bronze	2024-08-14
Simone	Biles	1997-03-14	oro	2024-09-01
Andre	De Grasse	1994-11-10	bronze	2024-08-10
Penny	Oleksiak	2000-06-13	oro	2024-09-11
Naomi	Osaka	1997-10-16	oro	2024-08-10
Hanyu	Yuzuru	1994-12-07	oro	2024-08-22
Neeraj	Chopra	1997-12-24	oro	2024-08-28
P.V.	Sindhu	1995-07-05	bronze	2024-08-16

8 row(s) fetched.

13. Queremos añadir el sufijo **" - Doble"** al apellido de los atletas que han ganado más de una medalla **(RA4 2 puntos)**

apellido
Makhloufi - Doble



Fraser-Pryce - Doble|  
Xiang - Doble |

3 row(s) fetched.

14. Actualizar el nombre del país añadiendo tantos \* al nombre como medallas de oro ha obtenido. (RA4 2 puntos).

id_pais	nombre	codigo_pais	continente
1	Nigeria	NGA	África
2	Kenia	KEN	África
3	Sudáfrica	ZAF	África
4	Egipto	EGY	África
5	Ghana	GHA	África
6	Marruecos	MAR	África
7	Argelia	DZA	África
12	Argentina	ARG	América
19	Japón	JPN	Asia
21	Corea del Sur	KOR	Asia
22	Rusia	RUS	Asia
23	Irán	IRN	Asia
24	Tailandia	THA	Asia
25	Indonesia	IDN	Asia
26	Kazajistán	KAZ	Asia
27	Filipinas	PHL	Asia
31	Australia	AUS	Australia
32	Nueva Zelanda	NZL	Australia
33	Fiyi	FJI	Australia
34	Alemania	DEU	Europa
35	Francia	FRA	Europa
36	Reino Unido	GBR	Europa
37	Italia	ITA	Europa
38	Polonia	POL	Europa
40	Ucrania	UKR	Europa
41	Países Bajos	NLD	Europa
42	Suecia	SWE	Europa
43	Noruega	NOR	Europa
8	Estados Unidos*	USA	América
9	Brasil**	BRA	América
10	México*	MEX	América
11	Canadá**	CAN	América
13	Colombia**	COL	América
14	Chile**	CHL	América
15	Perú*	PER	América
16	Jamaica*	JAM	América
17	Cuba**	CUB	América
18	China**	CHN	Asia
20	India*	IND	Asia
39	España***	ESP	Europa

40 row(s) fetched.



### PLPGSQL (RA5 10 puntos)

15. Crear una función llamada **contar\_medallas\_pais**, que reciba el nombre de un país y devuelva el número total de medallas (oro, plata, bronce) que ha ganado. Si el país no existe o no tiene medallas, debe devolver 0. (RA5 1.25 puntos)

```
select * from contar_medallas_pais('españa')

contar_medallas_pais|
-----+
                     4|

SELECT * from contar_medallas_pais('aNTartida')

contar_medallas_pais|
-----+
                     0|
```

16. Crear un procedimiento llamado **registrar\_medalla**, que registre una nueva medalla en la tabla resultados, recibiendo **id\_atleta**, **id\_deporte**, **medalla** y **fecha\_evento**. Debe verificar que el atleta, deporte existan antes de insertar. Además, el tipo de medalla debe ser oro o plata o bronce, si al insertar no se cumple algo de lo antes dicho de debe generar una excepción. Los mensajes serían **Atleta xx no existe**, **Deporte xx no existe** y **Medalla xx no válida**. (RA5 1.25 puntos)

Ejemplo **CALL registrar\_medalla(900, 99, 'oro', '2024-07-27');**

17. Crear un procedimiento llamado **reasignar\_pais**, que reasigne todos los atletas de un país origen a un país destino, actualizando **id\_pais** en atletas. Debe verificar que ambos países existan y registrar el cambio en una tabla de auditoría. Debes crear una tabla auditoria país con este comando:

```
CREATE TABLE auditoria_pais (
    id_auditoria SERIAL PRIMARY KEY,
    id_atleta INTEGER,
    pais_origen INTEGER,
    pais_destino INTEGER,
    fecha_cambio TIMESTAMP DEFAULT CURRENT_TIMESTAMP );
```

Además, debe generar excepciones **País origen xx no existe** o **País destino xx no existe** según corresponda. Te recuerdo de usar un **begin** antes de ejecutar el procedimiento y un **rollback** al final. (RA5 1.25 puntos)

18. Crear un trigger que, al insertar o actualizar una medalla en resultados, actualice una tabla **estadisticas\_pais** con el conteo total de medallas por tipo (oro, plata, bronce) para el país correspondiente. Para ello crea la tabla siguiente:

```
CREATE TABLE estadisticas_pais (
    id_pais INTEGER PRIMARY KEY,
    oros INTEGER DEFAULT 0,
    platas INTEGER DEFAULT 0,
    bronces INTEGER DEFAULT 0 );
```

Controla el error de meter una medalla que no sea oro, plata o bronce con un mensaje de excepción **'Medalla incorrecta.'** Haz pruebas insertando un par de medallas en la tabla de resultados. (RA5 2.5 puntos)



Ejemplo:

**INSERT INTO** resultados (id\_atleta, id\_deporte, medalla, fecha\_evento) **VALUES** (341, 1, 'plata', '2024-07-28');

```
SELECT * FROM estadisticas_pais

id_pais|oros|platas|bronces|
-----+-----+-----+-----+
39|    3|    2|    0|
```

19. Queremos crear una regla en la tabla resultados que impida insertar más de una medalla del mismo tipo (oro, plata o bronce) para un mismo deporte en la misma fecha de evento. Si se intenta insertar una medalla duplicada (por ejemplo, una segunda medalla de oro para el mismo deporte y fecha), la inserción debe ser redirigida a una tabla **medallas\_rechazadas** en lugar de añadirse a resultados. Esto simula que cada evento en un deporte solo puede tener un ganador por tipo de medalla por día. Debéis crear la tabla:

```
CREATE TABLE medallas_rechazadas (
  id_rechazo SERIAL PRIMARY KEY,
  id_atleta INTEGER,
  id_deporte INTEGER,
  medalla TEXT,
  fecha_evento DATE,
  motivo TEXT,
  fecha_rechazo TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Pista usar el **Where** de la sentencia rule. (RA5 2 puntos)

20. Crear una función llamada **listar\_medallistas\_pais** que reciba el nombre de un país y devuelva una tabla con los nombres completos de los atletas de ese país que han ganado una medalla (oro, plata o bronce), junto con el total de medallas que cada uno ha ganado. La función debe usar un cursor estático para recorrer los datos y devolver los resultados. Si el país no existe o no tiene medallistas, la función debe devolver una tabla vacía. Fijaos en la salida que se os aporta (RA5 1.75 puntos)

```
select * from listar_medallistas_pais('Antártida')

nombre_completo|total_medallas|
-----+-----+

select * from listar_medallistas_pais('Jamaica')

nombre_completo      |total_medallas|
-----+-----+
Shelly-Ann Fraser-Pryce|          2|
Usain Bolt            |          1|
Elaine Thompson-Herah |          1|
```