

Proyecto Final

Integrantes:	Arcos Morales Ramón :	319541478
	Casarrubias Casarrubias Victor Manuel :	421003581
	Castillo Hernández Eduardo :	420003557
	López Asano Miguel Akira :	320219089
	Rivera Lara Sandra Valeria :	320039823

Fecha: 29/11/2024

Índice

1. Índice de Archivos	2
2. Modelo Entidad-Relación	3
2.1. Justificación	6
3. Modelo Relacional	7
3.1. Dependencias Funcionales.	9
4. Lenguaje para Definición de Datos	11
5. Procedimientos Almacenados, Funciones y Disparadores	14
5.1. Procedimiento 1	14
5.2. Procedimiento 2	15
5.3. Procedimiento 3	16
5.4. Disparador 1	17
5.5. Disparador 2	18
5.6. Disparadores para la creación de Agenda (Disparadores 3-8)	19
6. Población de la Base de Datos	22
7. Consultas	26
7.1. Consulta 1	26
7.2. Consulta 2	27
7.3. Consulta 3	28
7.4. Consulta 4	29
7.5. Consulta 5	30
7.6. Consulta 6	31
7.7. Consulta 7	32
7.8. Consulta 8	33
7.9. Consulta 9	34
7.10. Consulta 10	35
7.11. Consulta 11	36
7.12. Consulta 12	37
7.13. Consulta 13	38
7.14. Consulta 14	39
7.15. Consulta 15	40

1. Índice de Archivos

El proyecto consta de los siguientes archivos:

```
. └── Diagramas
      ├── ERequipo.drawio
      ├── ERequipo.png
      ├── Relacionalequipo.drawio
      └── Relacionalequipo.png
└── Docs
    ├── Diccionario_equipo.pdf
    ├── ProyectoFinal_equipo.pdf
    └── README_equipo.pdf
└── SQL
    ├── DDL.sql
    ├── DML.sql
    ├── Query.sql
    └── fun.sql
```

4 directories, 11 files

Figura 1: Organización del proyecto

- **ERequipo.drawio/png:** Contiene el Modelo Entidad-Relación.
- **Relacionalequipo.drawio/png:** Contiene el Modelo Relacional
- **Diccionario_equipo.pdf:** Contiene el diccionario de datos de la base de datos.
- **ProyectoFinal_equipo.pdf:** Archivo actual.
- **README_equipo.pdf:** Readme del proyecto, incluye información de los integrantes.
- **DDL.sql:** Archivo con el script para construir la base de datos usando el Lenguaje para Definición de Datos.
- **DML.sql:** Archivo con el script para el poblamiento de la base de datos.
- **Query.sql:** Archivo que contiene las consultas en sql.
- **fun.sql:** Archivo con los procedimientos almacenados y disparadores.

2. Modelo Entidad-Relación

Para el modelo entidad relacional, se crearon las siguientes entidades con sus respectivos atributos:

- Persona (herencia total)
 - Nombre completo (nombre, primer apellido, segundo apellido)
 - Nacionalidad
 - Fecha de Nacimiento
 - Teléfono (multivaluado)
 - Correo (multivaluado)
 - Número Pasaporte (llave: Se tiene que el número de pasaporte es diferente para todo atleta, además es estándar para todo pasaporte sin importar nacionalidad.)
 - Atleta
 - Género
 - Juez/Árbitro
 - Entrenador
- Disciplina
 - Nombre (llave: Se tiene que cada nombre de las disciplinas es diferente)
 - Categoría
 - Número de participantes (Calculado)
- Patrocinador
 - Nombre (llave)
- Localidad
 - IdLocalidad (llave: Se usa una llave sustituta, pues la llave natural dirección resulta ser más complicada de manejar.)
 - Nombre
 - Dirección (calle, número, ciudad)
 - Tipo
 - Aforo
- Fase
 - Nombre (discriminante)
- País
 - Total de medallas (derivado)
 - Nombre (llave)
- Evento
 - IdEvento (llave: Al no encontrar una llave natural satisfactoria, se optó por una sustituta.)
 - Fecha
 - Hora de inicio
 - Duración máxima
- Entrada
 - Número de asiento
 - Folio (discriminante)
 - Costo Base

- Costo Final (derivado)
- Medalla
 - Número de Medalla (discriminante)
 - Tipo

Se cuenta con las siguientes relaciones:

- Supervisar de Juez a Disciplina. (Muchos a muchos con participación total ambos).
- Patrocinar de Patrocinador a Disciplina. (Muchos a muchos con participación total ambos).
- Enseñar de Entrenador a Disciplina. (Muchos a uno con participación total ambos).
- Practicar de Atleta a Disciplina. (Muchos a muchos con participación total ambos).
- Haber de Evento a Disciplina. (Muchos a uno con participación total en Evento).
- Referir de Medalla a Disciplina. Relación débil. (Muchos a uno con participación total ambos).
- Ganar de Atleta a Medalla. (Muchos a uno con participación total en medalla).
- Entrenar de Entrenador a Atleta. (Muchos a muchos con participación total ambos).
- Participar de Atleta a Evento. (Muchos a muchos con participación total ambos).
- Correspondiente de Evento a Fase. Relación débil (uno a uno con participación total ambos).
- Pertener de Fase a Entrada. Relación débil. (Muchos a uno con participación total ambos).
- Representar de Atleta a País. (Muchos a muchos con participación total ambos).
- Ubicar de País a Localidad. (Muchos a uno con participación total ambos).
- Tener de Evento a Localidad. (Muchos a uno con participación total en evento).

Con lo anterior, se tiene que el Modelo Entidad-Relación se observa en la figura 2.

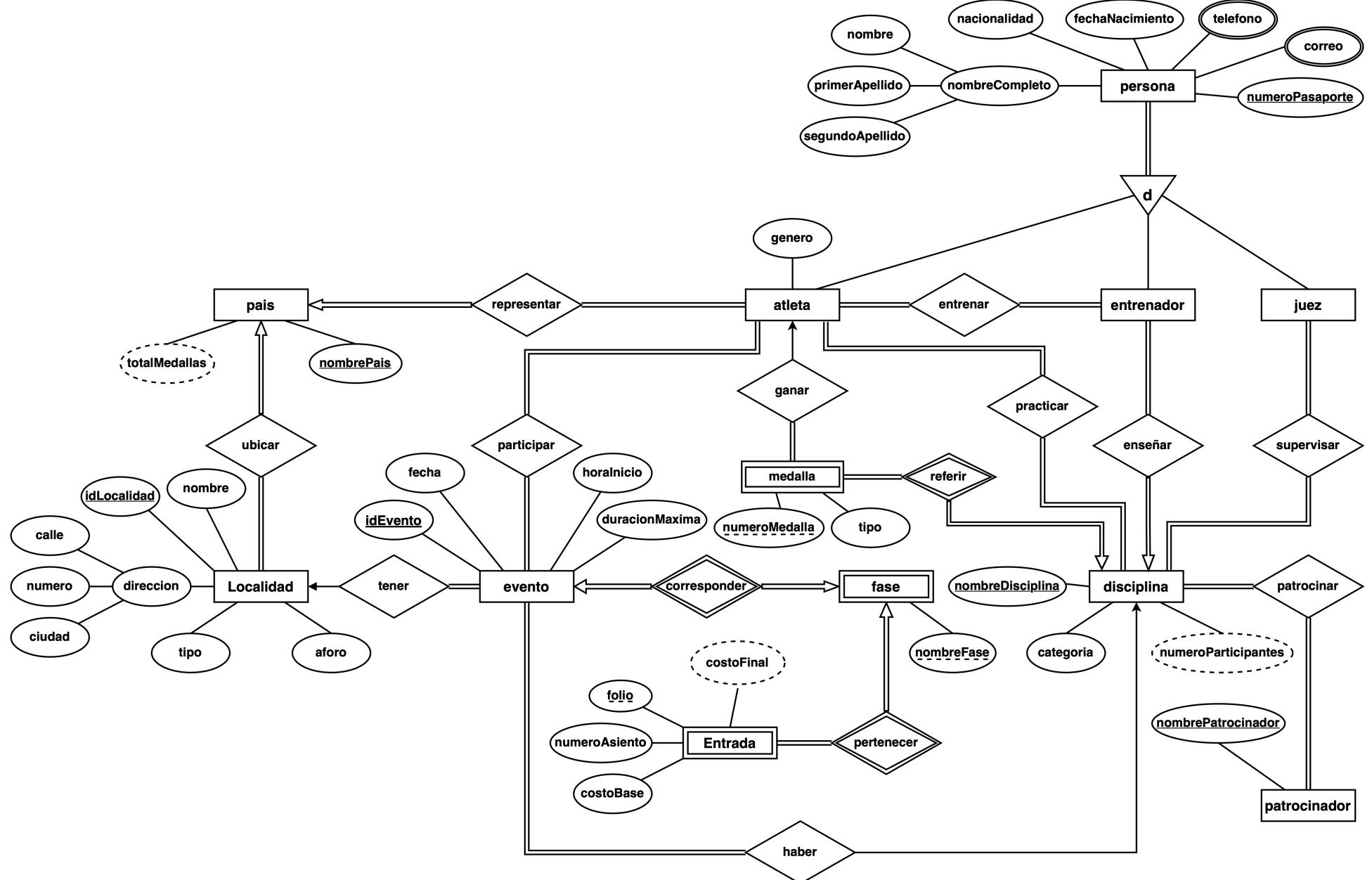


Figura 2: Modelo Entidad-Relación

2.1. Justificación

Para la elaboración del modelo Entidad-Relación, se tomaron las siguientes decisiones para empatar con la especificación proporcionada:

- Debido a que la entidad atleta, entrenador y juez comparten varios atributos como el nombre, la nacionalidad, teléfono o fecha de nacimiento, decidimos que exista una entidad persona para poder agrupar dichas entidades.
- Se decidió incluir el atributo de nacionalidad. Pues existen atletas que representan a algunos países diferentes de su nacionalidad.
- Consideramos que un atleta puede tener varios entrenadores ya que existen equipos de entrenadores.
- Una disciplina puede ser retirada de un evento si ya no se lleva a cabo.
- Se consideró que solo habrá un evento por fase eliminatoria. Es decir, no se tendrán un evento con finales y semifinales a la vez.
- Se decidió asignar el Número de pasaporte como llave a una persona. Ya que este es único y es requerido para poder participar en los juegos olímpicos. (Aunque sean nacionales)
- No se agregó la entidad agenda ya que esta puede ser consultada por medio de los eventos llevados a cabo.
- De igual manera, el medallero puede ser obtenido con una consulta sobre las medallas que ha ganado un país. Por lo tanto, se creó una entidad débil medalla para poder otorgar a un atleta.
- Se consideró la entidad Entrada como débil debido a que corresponde al boleto de la fase y este varía dependiendo de esta, se consideró Fase como débil debido a que cada fase pertenece a un evento; es decir, la fase no existe sin el evento.
- Localidad se encuentra en la relación pertenecer hacia un país, puesto que el país sede contiene todas las localidades.
- La llave de Patrocinador es nombre debido a que los nombres de los patrocinadores son únicos.
- Se tiene que un juez puede supervisar más de una disciplina si estas están estrechamente relacionadas.
- En la entidad Entrada el costo es calculado dependiendo de la fase en la que se encuentre.
- El atributo total de medallas de un país es calculado dependiendo de la cantidad de medallas que ganaron los atletas que lo representan.
- Se tiene que todo Patrocinador debe de patrocinar una Disciplina.

3. Modelo Relacional

Para la elaboración del Modelo Relacional, se realizó la traducción del Modelo Entidad-Relación a la siguiente forma reducida:

Juez(NúmeroPasaporte(PK), FechaNacimiento, Nacionalidad, Nombre, PrimerApellido, SegundoApellido)

Atleta(NúmeroPasaporte(PK), FechaNacimiento, Nacionalidad, Nombre, PrimerApellido, SegundoApellido, Género, NombrePaís(FK))

Entrenador(NúmeroPasaporte(PK), FechaNacimiento, Nacionalidad, Nombre, PrimerApellido, SegundoApellido, NombreDisciplina(FK))

País(NombrePaís (FK))

Localidad(IdLocalidad(PK), Nombre,Aforo, Tipo, Calle, Número, Ciudad, NombrePaís(FK))

Evento(IdEvento(PK), fecha, Hora inicio, Duración Máxima, IdLocalidad(FK), NombreDisciplina(FK))

Disciplina(NombreDisciplina(PK), Categoría)

Patrocinador(NombrePatrocinador(PK))

Fase(NombreFase(PK), IdEvento(PK))

Entrada(IdEvento(PK), NombreFase(PK), Folio(PK))

Medalla(NúmeroMedalla(PK), NombreDisciplina(PK), Tipo, NúmeroPasaporte(FK))

Participar(IdEvento(FK), NúmeroPasaporte(FK))

Entrenar(NúmeroPasaporteE(FK),NúmeroPasaporteA(FK))

Practicar(NúmeroPasaporte(FK),NombreDisciplina(FK))

Supervisar(NúmeroPasaporte(FK),NombreDisciplina(FK))

TeléfonoAtleta(NúmeroPasaporte(PK), Teléfono(PK))

CorreoAtleta(NúmeroPasaporte(PK), Correo(PK))

TeléfonoEntrenador(NúmeroPasaporte(PK), Teléfono(PK))

CorreoEntrenador(NúmeroPasaporte(PK), Correo(PK))

TeléfonoJuez(NúmeroPasaporte(PK), Teléfono(PK))

CorreoJuez(NúmeroPasaporte(PK), Correo(PK))

Con lo que el Modelo Relacional se observa en la figura 3.

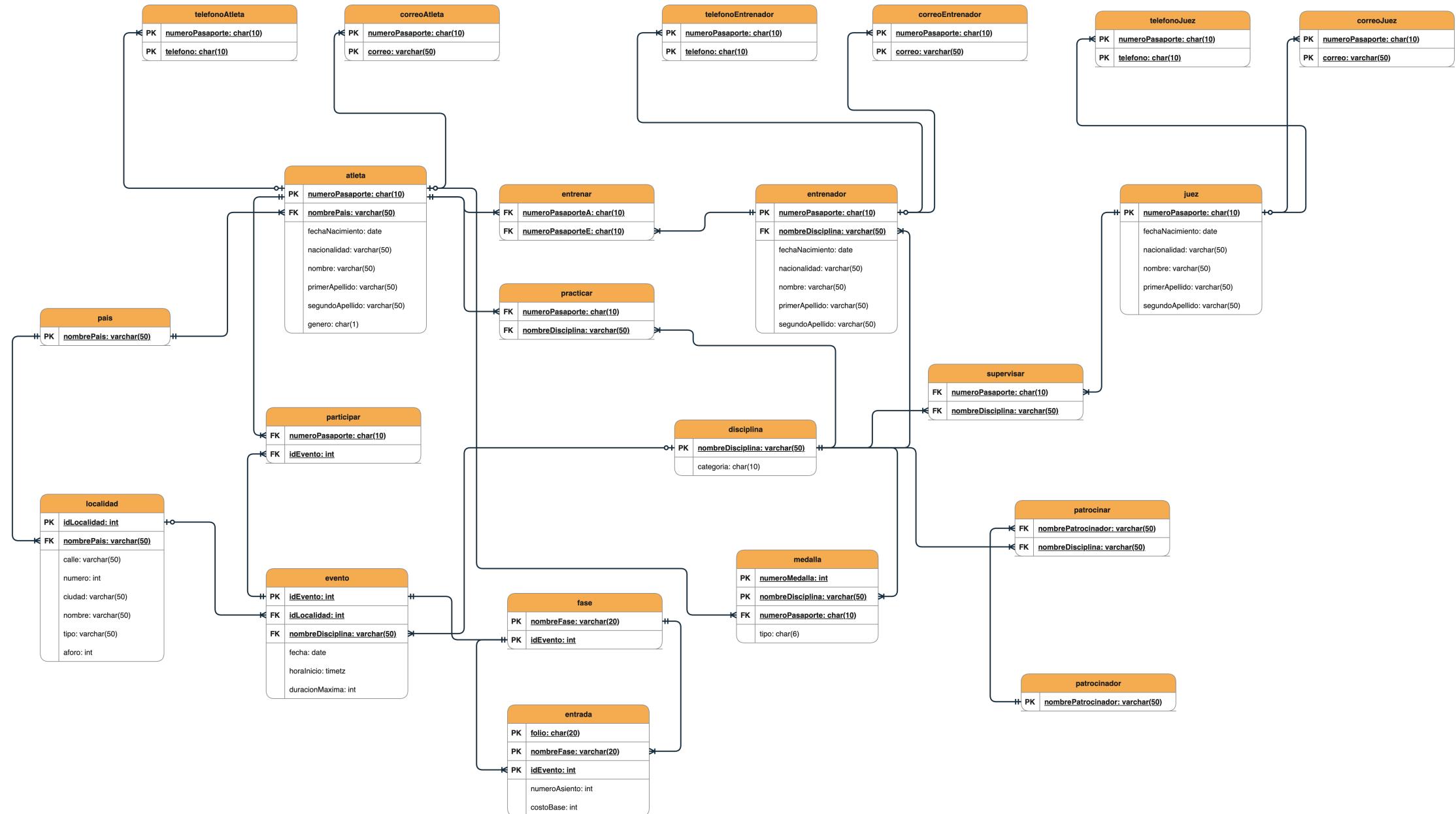


Figura 3: Modelo Relacional

3.1. Dependencias Funcionales.

Dado el Modelo Relacional, tiene el siguiente conjunto de dependencias funcionales:

Dependencias Funcionales de atleta:

- numeroPasaporte → nombrePais, fechaNacimiento, nacionalidad, nombre, primerApellido, segundoApellido, genero.

Dependencias Funcionales de telefonoAtleta:

- numeroPasaporte, telefono → numeroPasaporte, telefono

Dependencias Funcionales de correoAtleta:

- numeroPasaporte, correo → numeroPasaporte, correo

Dependencias Funcionales de entrenador:

- numeroPasaporte → nombreDisciplina, fechaNacimiento, nacionalidad, nombre, primerApellido, segundoApellido.

Dependencias Funcionales de telefonoEntrenador:

- numeroPasaporte, telefono → numeroPasaporte, telefono

Dependencias Funcionales de correoEntrenador:

- numeroPasaporte, correo → numeroPasaporte, correo

Dependencias Funcionales de juez:

- numeroPasaporte → fechaNacimiento, nacionalidad, nombre, primerApellido, segundoApellido.

Dependencias Funcionales de telefonoJuez:

- numeroPasaporte, telefono → numeroPasaporte, telefono

Dependencias Funcionales de correoJuez:

- numeroPasaporte, correo → numeroPasaporte, correo

Dependencias Funcionales de entrenar:

- numeroPasaporteA, numeroPasaporteE → numeroPasaporteA, numeroPasaporteE

Dependencias Funcionales de practicar:

- numeroPasaporte, nombreDisciplina → numeroPasaporte, nombreDisciplina

Dependencias Funcionales de supervisa:

- numeroPasaporte, nombreDisciplina → numeroPasaporte, nombreDisciplina

Dependencias Funcionales de pais:

- nombrePais → nombrePais

Dependencias Funcionales de disciplina:

- nombreDisciplina → categoria

Dependencias Funcionales de participar:

- nombrePais → idEvento

Dependencias Funcionales de patrocinador:

- nombrePatrocinador → nombrePatrocinador

Dependencias Funcionales de patrocinar:

- nombrePatrocinador, nombreDisciplina → nombrePatrocinador, nombreDisciplina

Dependencias Funcionales de localidad:

- idLocalidad → nombrePais, calle, numero, ciudad, nombre, tipo, aforo

Dependencias Funcionales de evento:

- idEvento → idLocalidad, nombreDisciplina, fecha, horaInicio, duracionMaxima

Dependencias Funcionales de fase:

- nombreFase, idEvento → nombrefase, idEvento

Dependencias Funcionales de entrada:

- folio, nombreFase, idEvento → numeroAsiento, costoBase

Dependencias Funcionales de medalla:

- numeroMedalla, nombreDisciplina → numeroPasaporte, tipo

4. Lenguaje para Definición de Datos

Se crearon las siguientes restricciones de dominio por cada tabla:

atleta:

- numeroPasaporte se definió como llave primaria, además que debe tener una longitud de 10.
- nombrePais se definió como llave foránea, además de verifica que no sea vacía.
- nacionalidad se verifica que no sea vacía.
- fechaNacimiento se verifica que no sea null.
- nombre se verifica que no sea vacío.
- primerApellido se revisa que no sea vacío.
- género sea revisa que sea M o F.

telefonoAtleta:

- telefono se define como llave primaria y que sea compuesto por dígitos.
- numeroPasaporte se define como llave primaria y que exista en Atleta, es decir como llave foránea.

correoAtleta:

- correo se define como llave primaria y que tenga la estructura de un dominio válido.
- numeroPasaporte se define como llave primaria y que exista en Atleta, es decir como llave foránea.

evento:

- idEvento se definió como PRIMARYKEY y se le definió una restricción a la columna para que no pueda ser Null.
- idLocalidad se definió como FOREIGNKEY que hace referencia a Localidad. Adicionalmente agrega la restricción para que no sea Null.
- nombreDisciplina se definió como FOREIGNKEY que hace referencia a Disciplina. Adicionalmente, se agrega la restricción para que no sea Null o una cadena vacía.
- fecha tiene como restricción en su columna para que no sea Null.
- horaInicio tiene como restricción en su columna que no sea Null.
- duracionMáxima tiene como restricción no ser Null o tener una duración negativa o mayor a 24, es decir un evento no puede durar más de 24 horas.

entrada:

- folio se definió como PRIMARYKEY compuesta con nombreFase e idEvento y se le definió una restricción para que no sea Null y la longitud de cualquier cadena de dicha columna sea igual a 20.
- nombreFase se definió como PRIMARYKEY compuesta y también hace referencia a Fase. Adicionalmente agrega la restricción para que no sea Null o una cadena vacía.
- idEvento se definió como PRIMARYKEY compuesta y también hace referencia a Evento. Adicionalmente agrega la restricción para que no sea Null.
- numeroAsiento tiene como restricción en su columna para que no sea Null.
- costoBase tiene como restricción que en su columna para que no sea Null y que dicho int no pueda ser menor a 0.

pais:

- nombrePais se definió como PRIMARYKEY y se le definió una restricción a la columna para que no pueda ser Null o una cadena vacía.

localidad:

- idLocalidad se definió como PRIMARYKEY y se le definió una restricción a la columna para que no pueda ser Null.
- nombrePais se definió como FOREIGNKEY que hace referencia a pais. Adicionalmente, se agrega la restricción para que no sea Null o una cadena vacía.
- calle tiene como restricción no ser Null o una cadena vacía.
- numero, se agregó una restricción a la columna para que no sea Null.
- ciudad tiene como restricción no ser Null o una cadena vacía.
- nombre tiene como restricción no ser Null o una cadena vacía.
- tipo tiene como restricción no ser Null o una cadena vacía.
- aforo tiene como restricción no ser Null y tener que ser mayor a 0.

patrocinador:

- nombrePatrocinador se definió como PRIMARYKEY y tiene como restricción no ser Null o una cadena vacía.

disciplina:

- nombreDisciplina se definió como PRIMARYKEY y tiene como restricción no ser Null o una cadena vacía.
- categoria tiene como restricción solo poder ser Individual o Equipo.

medalla:

- numeroMedalla se definió como PRIMARYKEY compuesta y tiene como restricción no ser Null y tener que ser mayor a 0.
- nombreDisciplina se definió como PRIMARYKEY compuesta que hace referencia a Disciplina, tiene como restricción no ser Null o una cadena vacía.
- numeroPasaporte se definió como FOREIGNKEY que hace referencia a Atleta, tiene definida una restricción sobre la columna para no ser Null.
- tipo tiene como restricción solo poder ser Oro o Plata o Bronce.

juez

- numeroPasaporte se definió como PRIMARYKEY y tiene la restricción de ser exactamente de longitud de 10 caracteres.
- fechaNacimiento se verifica que no sea nulo.
- nacionalidad se verifica que no sea vacío.
- nombre se verifica que no sea vacío.
- primerApellido se verifica que no sea vacío.
- segundoApellido se verifica que no sea vacío.

telefonoJuez

- numeroPasaporte se definió como PRIMARYKEY compuesta, que a su vez hace referencia al numeroPasaporte de Juez.
- telefono se definió como PRIMARYKEY compuesta y se compone de dígitos.

correoJuez

- numeroPasaporte se definió como PRIMARYKEY compuesta, que a su vez hace referencia al numeroPasaporte de Juez.
- correo se definió como PRIMARYKEY compuesta y a su vez cumple con la estructura de un correo electrónico válido.

fase

- nombreFase se definió como PRIMARYKEY compuesta y se verifica que no sea vacía.

- idEvento se definió como PRIMARYKEY compuesta y a su vez hace referencia a idEvento de Evento.

practicar:

- numeroPasaporte se define como llave foránea de atleta.
- nombreDisciplina se define como llave foránea de disciplina.

patrocinar:

- nombrePatrocinador se definió como FOREIGNKEY que hace referencia a Disciplina, tiene como restricción no ser Null o una cadena vacía.
- nombreDisciplina se definió como FOREIGNKEY que hace referencia a Disciplina, tiene como restricción no ser Null o una cadena vacía.

participar:

- numeroPasaporte se definió como FOREIGNKEY que hace referencia a Atleta. Adicionalmente se agrega la restricción para que no sea Null.
- idEvento se definió como FOREIGNKEY que hace referencia a Evento. Adicionalmente se agrega la restricción para que no sea Null.

supervisar

- numeroPasaporte se definió como FOREIGNKEY que hace referencia a numeroPasaporte de Juez.
- nombreDisciplina se definió como FOREIGNKEY que hace referencia a nombreDisciplina de Disciplina.

entrenador

- numeroPasaporte se definió como llave primaria, además que debe tener una longitud de 10.
- nombrePais se definió como llave foránea, además de verifica que no sea vacía ni nula.
- nacionalidad se verifica que no sea vacía.
- fechaNacimiento se verifica que no sea null.
- nombre se verifica que no sea vacío.
- primerApellido se revisa que no sea vacío.

telefonoEntrenador

- telefono se define como llave primaria y que sea compuesto por dígitos.
- numeroPasaporte se define como llave primaria y que exista en Entrenador, es decir como llave foránea.

correoEntrenador

- correo se define como llave primaria y que tenga la estructura de un dominio válido.
- numeroPasaporte se define como llave primaria y que exista en Entrenador, es decir como llave foránea.

entrenar

- Se verifica que numeroPasaporteA no sea nulo y se define como llave foránea que hace referencia a número pasaporte de Atleta.
- Se verifica que numeroPasaporteE no sea nulo y se define como llave foránea que hace referencia a número pasaporte de Entrenador.

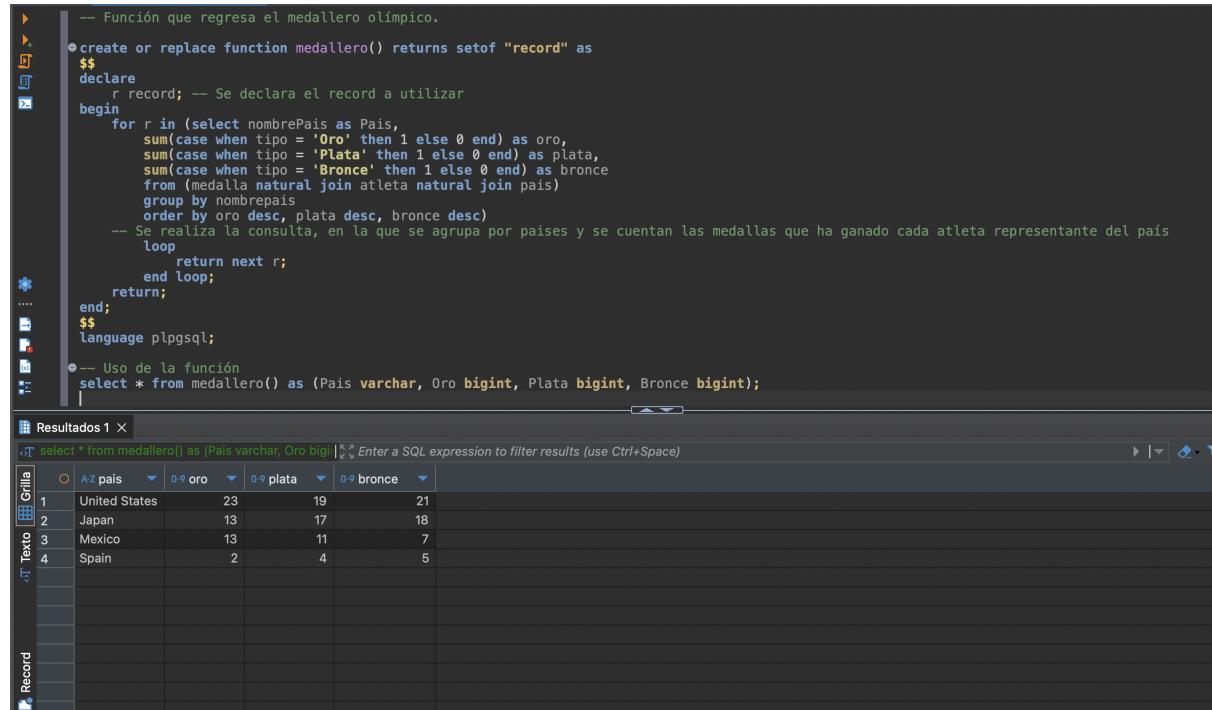
Se utilizó la política para mantenimiento de llaves de cascada tanto para el borrado como la actualización. Ya que permite mantener la integridad de datos de manera sencilla. Así, al borrar la llave se elimina la tupla con la llave foránea sin necesidad de eliminarla manualmente como sí sucede por omisión y tiene la ventaja de mantener los valores actualizados a diferencia de las políticas nula y por defecto. Con ello, al tener varias relaciones con llaves foráneas lo más sencillo es aplicar la política de cascada.

5. Procedimientos Almacenados, Funciones y Disparadores

5.1. Procedimiento 1

Se tiene una función llamada `medallero()` la cual no recibe parámetros y regresa una tabla que contiene los países ordenados por medallas ganadas.

El objetivo de la función es devolver el medallero olímpico actual de la competición. Ordenado por número de medallas y valor de estas.



```
-- Función que regresa el medallero olímpico.

CREATE OR REPLACE FUNCTION medallero() RETURNS SETOF "record" AS
$$
DECLARE
    r record; -- Se declara el record a utilizar
BEGIN
    FOR r IN (SELECT nombrePais AS País,
                    SUM(CASE WHEN tipo = 'Oro' THEN 1 ELSE 0 END) AS oro,
                    SUM(CASE WHEN tipo = 'Plata' THEN 1 ELSE 0 END) AS plata,
                    SUM(CASE WHEN tipo = 'Bronce' THEN 1 ELSE 0 END) AS bronce
                 FROM (medalla NATURAL JOIN atleta) NATURAL JOIN país
                 GROUP BY nombrePais
                 ORDER BY oro DESC, plata DESC, bronce DESC)
        -- Se realiza la consulta, en la que se agrupa por países y se cuentan las medallas que ha ganado cada atleta representante del país
        LOOP
            RETURN NEXT r;
        END LOOP;
    RETURN;
END;
$$
LANGUAGE plpgsql;

-- Uso de la función
SELECT * FROM medallero() AS (País VARCHAR, Oro BIGINT, Plata BIGINT, Bronce BIGINT);
|
```

Resultados 1 ×

	AZ País	09 oro	09 plata	09 bronce
1	United States	23	19	21
2	Japan	13	17	18
3	Mexico	13	11	7
4	Spain	2	4	5

Figura 4: Función para obtener el medallero olímpico

5.2. Procedimiento 2

Se tiene la función `capacidadRestante(idLocalidadS integer default null)`, la cuál tiene un parámetro opcional y regresa el idLocalidad, nombre, idEvento, fecha y lugaresDisponibles de la localidad.

Su objetivo es regresar la capacidad restante de cada localidad tomando en cuenta los eventos que ocurren en ella. Su parámetro opcional recibe un idLocalidad del cuál se desea obtener el número de lugares restantes. Si dicho parámetro no es proporcionado, la función regresará el resultado para todas las localidades disponibles.

Con dicha función es posible obtener resultados más específicos como aquellos con mayor número de asientos disponibles, o aquellos que están a punto de llenarse.

```

-- función que regresa la capacidad restante de una localidad si se especifica su id, en otro caso regresa la capacidad restante de todas las localidades
create or replace function capacidadRestante(idLocalidadS integer default null)
returns table (idLocalidad int, nombre varchar(50), idEvento int, fecha date, lugaresDisponibles int) as $$
begin

    if idLocalidadS is not null then
        if not exists (select * from localidad l where l.idLocalidad = idLocalidadS) then
            raise exception 'idLocalidad inválido, la localidad con id % no existe', idLocalidadS;
        end if;
    end if;

    return query
    select l.idLocalidad, l.nombre, e.idEvento, e.fecha,
           l.aforo-(select count(folio):: integer
                      from entrada en
                     where en.idEvento = e.idEvento) as lugaresDisponibles
      from localidad l
     join evento e on l.idLocalidad = e.idLocalidad
     where idLocalidadS is null or l.idLocalidad = idLocalidadS;
end;
$$
language plpgsql

```

RESULTS 1 ×

SELECT * FROM capacidadRestante() | Enter a SQL expression to filter results (use Ctrl+Space)

O	0-9 idlocalidad	A-Z nombre	0-9 idevento	0-9 fecha	0-9 lugaresdisponibles
	1	Los Angeles Memorial Coliseum	1	2028-07-20	69,994
	2	BMO Stadium	2	2028-07-18	21,995
	3	Dedeaux Field	3	2028-07-21	19,995
	4	Galean Center	4	2028-07-20	10,295
	5	Los Angeles Convention Center	5	2028-07-23	4,995
	6	Los Angeles Convention Center	6	2028-07-19	6,995
	7	Los Angeles Convention Center	7	2028-07-27	7,995
	8	Los Angeles Convention Center	8	2028-07-14	8,995
	9	Crypto Arena	9	2028-07-20	17,995
	10	Peacock Theater	10	2028-07-27	6,995
	11	Grand Park	11	2028-07-18	4,995
	12	Sepulveda Basin	12	2028-07-21	2,995
	13	Sepulveda Basin	13	2028-07-27	7,995
	14	Sepulveda Basin	14	2028-07-28	14,995
	15	Dignity Health Sports Park	15	2028-07-19	9,995
	16	Dignity Health Sports Park	16	2028-07-25	14,995
	17	Dignity Health Sports Park	17	2028-07-14	29,995
	18	Velodrome Sports Center	18	2028-07-27	5,995
	19	Long Beach Arena	19	2028-07-16	11,993
	20	Belmont Veterans Memorial Coliseum	20	2028-07-28	5,995
	1	Los Angeles Memorial Coliseum	21	2028-07-25	69,995
	2	BMO Stadium	22	2028-07-14	21,995
	3	Dedeaux Field	23	2028-07-14	19,995
	4	Galean Center	24	2028-07-21	10,295

Figura 5: Función para el número de lugares disponibles

5.3. Procedimiento 3

Se tiene la función `equipo(npais varchar, ndisciplina varchar)`, la cual tiene dos parámetros obligatorios, que son el país y la disciplina del equipo que queremos.

Su objetivo es regresar los nombres y pasaportes de los atletas que participan en la disciplina dada para el país dado, así como sus entrenadores. Es por ello de devuelve una tabla con los atributos de `numeroPasaporte, nombre, primerApellido y segundoApellido` de los atletas y entrenadores. Si no existe el país, o la disciplina, o atletas y entrenadores que cumplan las condiciones, se devuelve una tabla vacía.

Con esta función es posible conocer los equipos en una disciplina del país que nos interese de forma rápida. Con esto cada país podrá identificar a sus representantes.

```

CREATE OR REPLACE FUNCTION equipo(npais VARCHAR, ndisciplina VARCHAR) RETURNS SETOF "record" AS
$$
DECLARE
    r record;
BEGIN
    FOR r IN (SELECT a.numeroPasaporte, a.nombre, a.primerApellido, a.segundoApellido,
                  e.numeroPasaporte, e.nombre, e.primerApellido, e.segundoApellido
                FROM atleta AS a NATURAL JOIN practicar AS p
                JOIN entrenar AS et ON et.numeroPasaporteA = a.numeroPasaporte
                JOIN entrenador e ON et.numeroPasaporteE = e.numeroPasaporte
                WHERE a.nombrePais = npais AND p.nombreDisciplina = ndisciplina)
    LOOP
        RETURN NEXT r;
    END LOOP;
    RETURN;
END;
$$
LANGUAGE plpgsql;
SELECT * FROM equipo('United States', 'Archery')
AS (Atleta CHAR(10), nombreA VARCHAR, primerapellidoA VARCHAR, segundoapellidoA VARCHAR,
    Entrenador CHAR(10), nombreE VARCHAR, primerapellidoE VARCHAR, segundoapellidoE VARCHAR);

```

Resultados 1 ×

	atleta	nombreA	primerapellidoA	segundoapellidoA	entrenador	nombreE	primerapellidoE	segundoapellidoE
1	03CAULVONP	Haily	Cuthill	Moxley	KIYSESFYT9	Cullen	Gircke	Tabard
2	5XAZD5ZAES	Aili	Jacketts	Deerr	GR92ZF6832	Priscilla	Garcia	Lehemann
3	7YAI23UK1V	Karylin	Lofty	Hacket	TSAZEA1QWI	Anna-diane	Bonas	Fearnall
4	BDV763BXZI	Bryn	Priddy	Digwood	JOVHMJU2VI	Andrew	Snoxill	Jablonski
5	I1P3BGAMYV	Alfred	Winspurr	Brilon	B408P3MBU0	Cortney	McVitty	Poulgreen
6	ME4BZFFCS7	Skell	Vardy	Hagston	TY0IJPG0RK	Molli	Baldetti	Sirman
7	NYYCD1EH52	Johann	Addlestone	Syms	5SIB8OUHBU	Malva	Jouning	Dimelow
8	WESHIINACP	Gaspar	Bearward	Scorton	BR2C7WO9Y4	Carmon	Stedell	Deeth
9	Y4GK05XLHL	Jeremy	Lusk	Thoms	1YYJWIXQ5X	Allianora	Isoldi	Suff

Figura 6: Función para obtener los participantes de un país en una disciplina

5.4. Disparador 1

Disparador que revisa que no se exceda el aforo de la localidad, regresando una excepción si sucede el caso anterior.

Se pretende mantener la consistencia de los datos almacenados en la base de datos. Pues no se pueden tener más entradas que las indicadas por el aforo de cada localidad.

Como nota importante, esta función hace uso de la función que definimos previamente llamada *capacidadRestante(idLocalidad integer default null)* que nos regresa la capacidad restante, por lo que es necesario crearla antes de crear el disparador, además, es necesario ejecutar el disparador antes de ejecutar el archivo DML o cualquier otra inserción.

```

-- Trigger que revisa que no se exceda el aforo de la localidad.
create or replace function revisarDisponibilidad()
returns trigger as $$ 
declare
    targetLocalidad int;
    capacidadRestante int;
begin
    select idLocalidad into targetLocalidad
    from evento
    where idEvento = new.idEvento;

    select lugaresDisponibles
    into capacidadRestante
    from capacidadRestante(targetLocalidad)
    where idEvento = new.idEvento;

    if capacidadRestante <= 0 then
        raise exception 'Se ha alcanzado la capacidad máxima de la localidad especificada.';
    end if;

    return new;
end
$$ language plpgsql;

create trigger checkDisponibilidadTrigger
before insert or update on entrada
for each row
execute function revisarDisponibilidad();

```

⚠ SQL Error [P0001]: ERROR: Se ha alcanzado la capacidad máxima de la localidad especificada.
Where: PL/pgSQL function revisardisponibilidad()
line 17 at RAISE

insert into entrada (folio, nombreFase, idEvento, numeroAsiento, costoBase) values ('ZQ87-ifjd5glhjr-0009', 'Fase

Figura 7: Disparador para revisar capacidades

5.5. Disparador 2

Disparador que calcula el costo final de una entrada al insertar o actualizar una entrada a la base de datos.

Debido a que el costo final es un atributo calculado el cuál no es incluido a la hora de traducir al modelo relacional, este es añadido posterior a la creación de el DDL Se utilizó un alter table para agregar la columna costo final. Este campo es llenado por medio del disparador dependiendo de la fase actual y el costo base de la entrada. Se tiene especificado que el precio aumente en un 9 % por cada fase que se avance. En este caso, fase 4 es la última fase.

Como nota importante, es necesario definir y ejecutar el disparador antes de realizar las inserciones, pues de otra forma, los datos que ya están en la base no se ven afectados.

```
--DISPARADOR 2
-- Agregamos la columna costoFinal a la tabla entrada, al ser un atributo calculado, este no aparece en el modelo relacional.
alter table entrada add column costoFinal int;

-- Función que regresa un disparador, el cual se encarga de calcular el precio final de una entrada en específico.
create or replace function precioFinal()
returns trigger as $$
begin
    -- El precio aumentará en 9% en cada fase que se avance.
    new.costoFinal := new.costoBase *
        case
            when f.nombreFase = 'Fase 1' THEN 1.0 -- El precio es el mismo que el de base
            when f.nombreFase = 'Fase 2' THEN 1.09
            when f.nombreFase = 'Fase 3' THEN 1.18
            when f.nombreFase = 'Fase 4' THEN 1.27
        end
    from fase f
    where f.idEvento = new.idEvento;

    return new;
end;
$$ language plpgsql;
-- Asociamos el trigger a la tabla entrada cuando se decida insertar o actualizar.
create trigger actualizarPrecioFinal
before insert or update on entrada
for each row
execute function precioFinal();
```

Figura 8: Disparador encargado de la columna de precio final

	folio	nombrefase	idevento	numeroasiento	costobase	costofinal
1	ZQ87-ifjd5glhjr-5387	Fase 1	1	283	1,219	1,219
2	PW82-v4wzc842qy-5533	Fase 1	2	304	539	539
3	ZU40-mrbaiu8byd-5721	Fase 1	3	292	902	902
4	LC92-teoqzkdjfe-6445	Fase 1	4	951	1,300	1,300
5	DB12-9sfzdxyt-8223	Fase 1	5	494	1,488	1,488
6	HF65-c2822326sr-0379	Fase 1	6	348	1,391	1,391
7	FA42-3gehf6w5y6-2400	Fase 1	7	805	2,807	2,807
8	YM96-adj0jovzc5-0010	Fase 1	8	369	942	942
9	GH49-4emclzfcm-8948	Fase 1	9	324	2,937	2,937
10	AL82-rgv3wzoei-2846	Fase 1	10	515	1,692	1,692
11	FM65-4ehcyub2qi-6630	Fase 1	11	174	1,264	1,264
12	SG76-u8c31zosd9-6582	Fase 1	12	148	2,434	2,434
13	CM19-8tqpoou592-7408	Fase 1	13	312	2,589	2,589
14	WH87-ogku2dnyxw-3191	Fase 1	14	175	1,208	1,208
15	BT42-0epcr46xxl-7379	Fase 1	15	392	1,405	1,405
16	NY14-qnjnq2286n-4564	Fase 1	16	210	686	686
17	MI10-5ccsep9mou-2862	Fase 1	17	216	2,909	2,909
18	JH05-qt6cgtpo7-3704	Fase 1	18	485	2,841	2,841
19	QK38-4e6rokyp4-1726	Fase 1	19	845	2,468	2,468
20	OA04-owwpfcnsne-0721	Fase 1	20	122	2,998	2,998
21	IQ76-q998ih3gg-3009	Fase 1	21	342	2,817	2,817
22	DL48-d8ndrqzfan-3785	Fase 1	22	466	1,536	1,536
23	CB07-b2ojo41x4r-9645	Fase 1	23	796	2,088	2,088
24	RB14-akjh0f00mk-2412	Fase 1	24	510	1,634	1,634
25	WF58-ddh7an2nna-2832	Fase 1	25	227	2,944	2,944
26	JC06-inbgn8mmma-4086	Fase 1	26	253	516	516
27	DC43-lww6evqdra-1036	Fase 1	27	495	2,971	2,971
28	PZ82-hmdidzvc18-0469	Fase 1	28	156	1,509	1,509
29	WZ87-id3wnwgqw3-3029	Fase 1	29	778	2,068	2,068
30	VK11-zol01lsbd9-4286	Fase 1	30	329	1,771	1,771
31	WG73-p778mb02nj-1669	Fase 1	31	956	2,893	2,893
32	UX60-499t6wwwd3-3463	Fase 1	32	204	1,686	1,686
33	TH77-bn63izi7eu-4683	Fase 1	33	150	1,581	1,581

Figura 9: Tabla evento

5.6. Disparadores para la creación de Agenda (Disparadores 3-8)

Debido a que se requiere de una agenda que contenga los distintos eventos y la información relacionada a ella como el precio, participantes, nombre, entre otros. Se decidió manejar la creación de la agenda por medio de disparadores. Pues esta es creada con la información de distintas tablas y se requiere que dicha agenda contenga la información actualizada.

Primero se define la tabla agenda con los atributos requeridos. Además, se le define su llave primaria y las llaves foráneas. Posteriormente, se definen 6 disparadores que se encargarán de manejar los distintos casos que pueden ocurrir para poder mantener la tabla agenda con la información solicitada.

El precio se decidió tomar el más bajo, pues se decidió tomar dicho campo como el "precio desde".

Disparador 3: Creacion/Actualización de un evento

Este disparador se encarga de realizar una inserción dentro de la tabla agenda en cuanto se detecte una inserción o actualización en la tabla evento. Este disparador toma la información sobre evento, entrada y participar. Si el evento ha sido definido y no se tienen participantes o entradas, el campo correspondiente tomará el valor nulo hasta que suceda un cambio en dichas tablas.

```

--Creamos la tabla de agendas, la cuál contiene la información de los eventos y su respectiva información
create table agenda(
    idEvento int,
    idLocalidad int,
    nombreDisciplina varchar(50),
    fecha date,
    horaInicio timetz,
    duracionMaxima int,
    precio int,
    participantes text
);
--Llave primaria
alter table agenda add constraint agenda_pk primary key (idEvento);
--Llaves foráneas
alter table agenda add constraint agenda_fk1 foreign key (idLocalidad) references localidad(idLocalidad) on delete cascade on update cascade;
alter table agenda add constraint agenda_fk2 foreign key (nombreDisciplina) references disciplina(nombreDisciplina) on delete cascade on update cascade;
--DISPARADOR 3 creacion/ actualización de evento.
--Definimos la función que ejecutará el trigger, esta se encarga de juntar toda la información.
create function creacionEventosAgenda()
returns trigger as $$ 
declare
    participantes text;
    precio int;
begin
    --Si no hay entradas, este será nulo
    select e.costoFinal
    into precio
    from entrada e
    where e.idEvento = new.idEvento;
    --Si no hay participantes, la entrada será nula
    select string_agg(e.numeroPasaporte, ', ')
    into participantes
    from participar p
    where p.idEvento = new.idEvento;
    insert into agenda(idEvento, idLocalidad, nombreDisciplina, fecha, horaInicio, duracionMaxima, precio, participantes)
    values (new.idEvento, new.idLocalidad, new.nombreDisciplina, new.fecha, new.horaInicio, new.duracionMaxima, new.precio, new.participantes);
    return new;
end;
$$ language plpgsql;
--Creamos el trigger y lo asignamos a evento
create trigger crearEventoAgenda
after insert or update on evento
for each row
execute function creacionEventosAgenda();

```

Figura 10: Definición de agenda y Disparador 3

Disparador 4: Creacion/Actualización de entradas

Este disparador se encarga realizar un update dentro de la fila correspondiente en agenda, utilizando el idEvento de la entrada creada(o actualizada). Además, el disparador siempre toma el precio más bajo de las entradas que se tienen asociadas al evento, pues se decidió tomar el precio desde.

Disparador 5: Creacion/Actualización de participantes

Este disparador se encarga realizar un update dentro de la fila correspondiente en agenda, utilizando el idEvento que es asociado al atleta en la tabla participar. Debido a que se pueden tener varios atletas participando en un evento, y solo se desea conocer a los atletas. Se decidió utilizar una cadena para guardar al identificador de los atletas. Con esto, se busca facilitar el manejo y lectura de la tabla agenda.

Disparador 6: Eliminar evento

Este disparador es encargado de manejar el caso en donde se elimine un evento y por lo tanto, este ya no esté disponible. En dicho caso, el disparador se encarga de eliminar la fila correspondiente en la tabla agenda.

Disparador 7: Eliminar entrada Este disparador se encarga de manejar el caso en donde una entrada es eliminada. Que este caso suceda, implica que el precio que se tiene en agenda puede ser afectado. Por lo tanto, este se encarga de actualizar el precio dentro de agenda con el mínimo de los que se tienen asociados al evento. En dado caso de que se eliminen todas las entradas asociadas, el valor se regresará a ser nulo.

```

--DISPARADOR 4 creacion/actualización de entradas.
--Función que ejecuta el disparador, este actualiza el valor de costo final cuando una entrada sea insertada o eliminada. Se quedará el costo más pequeño. Pues consideraremos el precio como precio desde...
create or replace function creacionEntradasAgenda()
returns trigger as $$

declare
    precioNuevo int;
begin
    --Seleccionamos el mínimo, si no hay más entradas, se asignará el valor null
    select min(e.costoFinal)
    into precioNuevo
    from entrada e
    where e.idEvento = new.idEvento;

    update agenda
    set precio = precioNuevo
    where idEvento = new.idEvento;
    return new;
end;
$$ language plpgsql;
-- Creamos el trigger y lo asignamos a entrada
create trigger creacionEntradasAgenda
after insert or delete on entrada
for each row
execute function creacionEntradasAgenda();

--DISPARADOR 5 creacion/actualización de participantes.
--Función que ejecuta el disparador, esta actualiza la tabla de participar.
create or replace function creacionParticipaAgenda()
returns trigger as $$

declare
    participantesNuevos text;
begin
    --Agregamos los nuevos participantes a nuestra cadena.
    select string_agg(p.numeroPasaporte, ', ')
    into participantesNuevos
    from participante p
    where p.idEvento = new.idEvento;

    update agenda
    set participantes = participantesNuevos
    where idEvento = new.idEvento;
    return new;
end;
$$ language plpgsql;
-- Creamos el trigger y lo asignamos a participar.
create trigger creacionParticipaAgenda
after insert or update on participante
for each row
execute function creacionParticipaAgenda();

--DISPARADOR 6 Eliminar evento
-- Función que ejecuta el disparador al eliminar un evento, esta función elimina la entrada de la agenda.
create or replace function eliminarEventoAgenda()
returns trigger as $$

begin
    delete from agenda
    where idEvento = old.idEvento;
    return old;
end;
$$ language plpgsql;
-- Creamos y asignamos el disparador al eliminar en evento.
create trigger eliminarEventoAgenda
after delete on evento
for each row
execute function eliminarEventoAgenda();

```

Figura 11: Disparadores 4,5,6

Disparador 8: Eliminar participación Este disparador se encarga de manejar el caso en donde una participación es eliminada. Si se elimina la participación de un atleta, esto implica que es necesario actualizar a los participantes de un evento. Para esto, se vuelven a obtener los participantes que se tienen asociados al evento. Por lo que la cadena se ve actualizada.

```

-- DISPARADOR 7 eliminar Entrada
-- Función que ejecuta el disparador al eliminar una entrada. La función se encarga de actualizar el costo al más bajo disponible dentro de las entradas actuales.
create or replace function eliminarEntradaAgenda()
returns trigger as $$

declare
    precioNuevo int;
begin
    --Seleccionamos el mínimo
    select min(e.costoFinal)
    into precioNuevo
    from entrada e
    where e.idEvento = old.idEvento;

    update agenda
    set precio = precioNuevo
    where idEvento = old.idEvento;
    return old;
end;
$$ language plpgsql;
-- Creamos y asignamos el disparador al eliminar en entrada.
create trigger eliminarEntradaAgenda
after delete on entrada
for each row
execute function eliminarEntradaAgenda();

-- DISPARADOR 8 Eliminar participante,
-- Función que ejecuta el disparador al eliminar un participante. La función actualiza la cadena con los numero de pasaporte de los participantes.
create or replace function eliminarParticipaAgenda()
returns trigger as $$

declare
    participantesNuevos text;
begin
    --Agregamos los participantes a nuestra cadena.
    select string_agg(p.numeroPasaporte, ', ')
    into participantesNuevos
    from participante p
    where p.idEvento = old.idEvento;

    update agenda
    set participantes = participantesNuevos
    where idEvento = old.idEvento;
    return old;
end;
$$ language plpgsql;
-- Creamos y asignamos el disparador al eliminar en participar.
create trigger eliminarParticipaAgenda
after delete on participante
for each row
execute function eliminarParticipaAgenda();

```

Figura 12: Disparadores 7 y 8

Finalmente, se tendrá la tabla agenda que contendrá la información de los eventos, precios y participantes de esta misma. Dicha tabla será consistente gracias a los 6 disparadores definidos.

	0	idevento	0	idlocalidad	AZ	nombrerdisciplina	0	fecha	0	horainicio	0	duracionmaxima	0	precio	AZ	participantes
1		106	6	6	Baseball	Softball	2028-07-23	12:54:00 -0600			2	1,044	1QKERMSTPN, 363ZNNN9W7, 4936VC93GI, BH6D60B6KR,			
2		112	12	6	Canoe	Slalom	2028-07-20	22:45:00 -0600			4	740	4GVAJQ94F, BPKGSNNDVE3, ICBT0V4DQZ, QJUFPWCE2B			
3		77	17	6	Hockey		2028-07-20	17:59:00 -0600			3	993	36C4RCBBFE, ANNH5KG7BN, H684D3QE9, OYEW1T18V8			
4		20	20	6	Diving		2028-07-28	18:58:00 -0600			4	1,912	0VBPT7RVXD, U8ME1956DT, FECRKHAS6O, MVOHD7LKLK			
5		84	4	6	Rowing		2028-07-20	21:54:00 -0600			1	1,578	3IKZ73DMLT, ATUOMNYNGAD, HKXZQ5M2LE, PA28Q5VPY,			
6		78	18	6	Judo		2028-07-14	22:54:00 -0600			4	592	3B02D71FL, APOLQU20XE, HG8PZ5Y0V, P2AYN5642Y, V			
7		22	2	6	Fencing		2028-07-14	07:09:00 -0600			5	679	0W3JP0WQZM, 4012MC19QV, B07I19TN25, FEMAI1OCJ0			
8		27	7	6	Hockey		2028-07-29	21:35:00 -0600			2	1,125	140EHJSASL, 7MBSK7MQOS, 8FPCDCSQQ, FGRT336BL9			
9		26	6	6	Handball		2028-07-23	07:32:00 -0600			5	516	10WUVXT5FY, 65Y5ZLLEHL, 8RTDLIZAU6, FGB2JLN04A, V			
10		79	19	6	Karate		2028-07-16	11:58:00 -0600			2	822	3BPUXB69W8Q, AR56KG8251, HDHKD14RN7, P2Z48S82G0,			
11		113	13	6	Canoe Sprint		2028-07-21	07:06:00 -0600			1	1,296	4IYCSU2CCO, BPKPUJC4XJG, IFQIMLMJ3Q, QKMR4YFV5F,			
12		46	6	6	Trampoline		2028-07-17	10:45:00 -0600			3	838	1SXZRS2NX9, 5H9GCPCTN7, 9KY3KHI634, G30C69J212,			
13		10	10	6	Boxing		2028-07-27	09:07:00 -0600			3	1,341	01LQFH4UXK, 45W6WM9Y0G, 89S86RG2CM, F56KX9T7			
14		25	5	6	Golf		2028-07-26	11:25:00 -0600			3	847	0ZU5MXTA7V, 27819LX79V, 8NVPNC3KEW, FG7J92BLR, N			
15		30	10	6	Lacrosse		2028-07-25	09:05:00 -0600			2	1,771	15LFDB31L4, 4ZJZ1I2AMA, 8WXEX1ATC3, FJRCRD16U3, N			
16		97	17	6	Triathlon		2028-07-19	19:07:00 -0600			5	1,135	10WUVXT5FY, 406CVD4E4Q, 6RM3U6FEQG, BY7YYKQGT1			
17		4	4	6	Athletics		2028-07-20	17:15:00 -0600			2	1,300	051HEFSZGT, 4C6ZG101ER, 81EG51Q404, FOJ5GRMILS, MG			
18		96	16	6	Trampoline		2028-07-24	09:12:00 -0600			3	1,003	3YFT78WAGW, BTALIJGM, HY2GXFDWAB, PN2XG7D0X,			
19		29	9	6	Karate		2028-07-18	23:46:00 -0600			2	1,963	14V2H6P10D, 8V8KG0B4P, FJ7TUUDX51, N49P69UR1J, L			
20		34	14	6	Rowing		2028-07-29	15:52:00 -0600			4	547	10GTPPRLS9, 93ZM0V9B19, FNHSTG4Y0, NC072BVEX3,			
21		55	15	6	Badminton		2028-07-19	07:58:00 -0600			1	713	2B0EMML13W, 50LT61Q6W, KALPHDJDYJ, 7XX95S1V5,			
22		98	18	6	Volleyball		2028-07-14	15:20:00 -0600			5	1,741	0R3V88A39T, 41J3HHWA5I, B95L9LGWY5, HZGJD37X0,			
23		56	16	6	Baseball Softball		2028-07-26	21:48:00 -0600			3	561	2D0N5J4FXD, 7UW9JN9ZQ, 9UB3GTN17A, GK25L0LUX4			
24		59	19	6	Beach Volleyball		2028-07-19	06:47:00 -0600			2	1,553	2GLU0H6BCB, 9X44N5HLL, GTD640FACD, 071DFHSVFF			
25		7	7	6	Basketball		2028-07-27	13:00:00 -0600			4	967	0AG89XNVQ, 4DLM12X0PX, 866126K885, F39VWH87K,			
26		80	20	6	Lacrosse		2028-07-23	12:07:00 -0600			1	1,194	3CQKNM44JQ, 76YZLW8EQ, ARBQ4U15B, H3EJV9GQ			
27		50	10	6	Weightlifting		2028-07-28	16:49:00 -0600			1	844	21M3MKNSKV, 9PXX0CHWD, GC3HJUE3D, NJUZBON3J			
28		37	17	6	Shooting		2028-07-27	05:30:00 -0600			3	2,327	1KOP1B01P, 0C915LJG3R, FOMC653VEU, NFH57P7G1, UB			
29		66	6	6	Cycling BMX Racing		2028-07-20	16:31:00 -0600			4	813	2MGZC6P1A, AK3D6185VK, H3G1N6K4U, OEPV5TSZK,			
30		69	9	6	Cycling Track		2028-07-28	12:39:00 -0600			1	561	25AKUAL64Y, 2QTWKRSV1, A9L9AVLFRV, H49TZ2MRU			
31		32	12	6	Modern Pentathlon		2028-07-29	11:59:00 -0600			4	865	1CD8WDMSCY, 91B86LMOQ, FKZXS9Z3BDJ, N7PMF9143C			
32		70	10	6	Diving		2028-07-16	08:04:00 -0600			2	784	2RTZXS9N5, 5VTDW2VTY, A97R7YCUWF, H5G5E05T2			
33		71	11	6	Equestrian		2028-07-29	06:52:00 -0600			5	1,078	2RAVCZTDX, AAMANRLM, H7XX3CRNCZ, OLSBQ5QF1N			

Figura 13: Tabla agenda

6. Población de la Base de Datos

La herramienta utilizada para la generación de datos fue Mockaroo. Se usaron los schemas que ofrece la pagina; es decir, una clase de template con la que se pueden poner atributos y datos que se requiere que generen. Por ejemplo, para elegir generar ids se usó un conjunto de datos y configuraciones definidos por Mockaroo como row numbers, para que se le asignara una serie de ids únicos a cada tupla. Para poder conectar los diferentes datos y schemas, se hizo uso de los datasets, la cuál es una forma de almacenar los datos generados de un schema por Mockaroo y guardarlos localmente para ser utilizados en otros schemas, como por ejemplo, para poder conectar los atributos que hacen referencia a tuplas de otras tablas, las cuales son llaves foráneas. Por último, para poder generar el código de inserción en SQL se hizo uso de los schemas de igual manera, ya que existe la opción de cambiar el formato de guardado local en SQL, que cambia automáticamente el código generado por Mockaroo de datos en CSV a SQL, luego con esos archivos, solo se guardó en el archivo DML para su uso en nuestra base de datos.

Descripto lo anterior, se muestran algunos ejemplos a la hora de poblar ciertas tablas:

país

Se generaron 3000 países y se aplicó *shell scripting* para ordenar y eliminar duplicados.
Resultado final, 180 países únicos.

El atributo nombrePais fue generado con el tipo *Country* de Mockaroo.

Relación con llave primaria.

disciplina

Se generó el CSV a mano de las 50 disciplinas oficiales según la página oficial de los juegos olímpicos de *Los Angeles 2028*.

Se generaron las 100 disciplinas restantes de la siguiente manera:

El atributo nombreDisciplina con el tipo de dato *Buzzword*

El atributo categoria con una expresión regular que escoge entre Individual o Equipo.

El CSV fue después convertido a SQL con una herramienta externa.

Relación con llave primaria.

atleta

Se generaron 1000 atletas.

Atletas representativos de Estados Unidos, Reino Unido, Australia y Canadá.

El atributo numeroPasaporte fue generado con una secuencia de 10 caracteres de dígitos aleatorios y letras en mayúsculas.

El atributo nombrePais fue generado con la restricción de los cuatro países antes mencionados.

El atributo fechaNacimiento fue considerado a partir del 1 de enero de 1980 hasta el 31 de diciembre del 2009.

El atributo nacionalidad fue generado con una distribución de peso con los siguientes valores: Estados Unidos 200, Canadá 50, Australia 40, Reino Unido 30, todos los demás países 1.

El atributo nombre fue generado con el tipo de dato *First Name*.

El atributo primerApellido fue generado con el tipo de dato *Last Name*.

El atributo segundoApellido fue generado con el tipo de dato *Last Name*.

El atributo género fue generado con el tipo de dato *Gender(abbrev)*.

Relación con llave primaria y foránea.

telefonoAtleta

Se generaron 1200 teléfonos.

Todos los atletas tienen al menos un teléfono y 200 tienen exactamente dos.

El atributo teléfono fue generado con el tipo de dato *Phone*.

Relación con llave compuesta.

correoAtleta

Se generaron 1000 correos.

Todos los atletas tienen al menos un correo y 200 tienen exactamente dos.

Relación con llave compuesta.

Se crearon los campos no visibles *_nombre*, *_usuario* y *_dominio*: el primero recupera el nombre del atleta, el segundo se genera con el tipo de dato *Username* y el último se genera con una lista personalizada con los valores *gmail*, *outlook*, *hotmail*, *yahoo*, *icloud* y una distribución de peso.

Con los datos anteriores se genera el atributo correo con la siguiente formula en *Ruby*

```
1 lower ("#{__nombre} . #{__usuario}@#{__dominio}.com")
```

entrenador

Se generaron 1000 entrenadores.

La construcción fue similar a los datos de la relación atleta, siendo la única diferencia que un entrenador puede haber nacido desde el 1 de enero de 1950.

Relación con llave primaria y foránea.

telefonoEntrenador

Se generaron 1200 teléfonos.

Todos los entrenadores tienen al menos un teléfono y 200 tienen exactamente dos.

El atributo teléfono fue generado con el tipo de dato *Phone*.

Relación con llave compuesta.

correoEntrenador

Se generaron 1000 correos.

Todos los entrenadores tienen al menos un correo y 200 tienen exactamente dos.

Relación con llave compuesta.

Contrucción análoga a correoAtleta.

juez

Se generaron 1000 jueces.

La construcción fue similar a los datos de la relación atleta, siendo la única diferencia que un juez puede haber nacido desde el 1 de enero de 1950.

Relación con llave primaria y foránea.

telefonoJuez

Se generaron 1200 teléfonos.

Todos los jueces tienen al menos un teléfono y 200 tienen exactamente dos.

El atributo teléfono fue generado con el tipo de dato *Phone*.

Relación con llave compuesta.

correoJuez

Se generaron 1000 correos.

Todos los entrenadores tienen al menos un correo y 200 tienen exactamente dos.

Relación con llave compuesta.

Contrucción análoga a correoAtleta.

patrocinador

Se generaron 210 patrocinadores y se aplicó *shell scripting* para ordenar y eliminar repetidos.

Resultado final, 200 patrocinadores únicos.

El atributo nombrePatrocinador fue generado con el tipo de dato *Fake Company Name*.

Relación con llave primaria.

localidad

Se generó el CSV a mano de 20 localidades oficiales.

Se generaron las 130 localidades de la siguiente manera:

El atributo idLocalidad con un *Row Number*.

El atributo nombrePais con un *Dataset Column* a la relación pais.

El atributo calle con un *Street Name*.

El atributo número con *Street Number*.

El atributo ciudad con *City*.

Los atributos nombre y tipo con *Buzzword*.

El atributo aforo con *Number*, teniendo un mínimo de 100 y un máximo de 10000.

Se agregó una localidad más para satisfacer ciertas consultas.

Resultado final 151 localidades.

El CSV fue después convertido a SQL con una herramienta externa.

Relación con llave primaria y llave foránea.

evento

Se generaron 200 eventos en el período oficial de los juegos olímpicos de *Los Angeles 2028*. Los eventos cubren el período que comprende del 14 de Julio de 2028 hasta el 30 de Julio de 2028. El atributo idEvento fue generado con el tipo de dato *Row Number*. Los atributos idLocalidad y nombreDisciplina fueron generados con el tipo de dato *Dataset Column* de manera secuencial. El atributo fecha fue generado con el tipo de dato *Datetime* con los períodos antes mencionados. El atributo horaInicio fue generado con el tipo de dato *Time* en formato 24 horas. El atributo duracionMaxima fue generado con el tipo de dato *Number* con un mínimo de 1 y un máximo de 5. Se agregaron 13 eventos para satisfacer ciertas consultas. Resultado final 213 eventos. Relación con llave primaria y llave foránea.

fase

Se generaron 200 fases. El atributo nombreFase fue generado con una expresión regular entre las cuatro fases. El atributo idEvento fue generado con el tipo de dato *Dataset Column* de manera secuencial. Se agregaron 13 fases para satisfacer ciertas consultas. Resultado final 213 fases. Relación con llave primaria y llave foránea.

entrada

Se generaron 1000 entradas. El atributo folio fue generado con el tipo de dato *Character Sequence* siguiendo el mismo patrón para todas las entradas. El atributo nombreFase fue generado con el tipo de dato *Dataset Column* de manera secuencial. El atributo idEvento fue generado con el tipo de dato *Dataset Column* de manera secuencial. El atributo numeroAsiento fue generado con el tipo de dato *Number* con un mínimo de 100y un máximo de 999. El atributo costoBase fue generado con el tipo de dato *Number* con un mínimo de 500y un máximo de 3000. Se agregaron 9 entradas para satisfacer ciertas consultas. Resultado final 1009 entradas. Relación con llave primaria y llave foránea.

medalla

Se generaron 150 medallas. El atributo numeroMedalla fue generado con el tipo de dato *Row Number*. El atributo nombreDisciplina fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación disciplina. El atributo numeroPasaporte fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación atleta. El atributo tipo fue generado con el tipo de dato *Custom List*. Se agregaron 3 medallas para satisfacer ciertas consultas. Resultado final 153 medallas. Relación con llave primaria y llave foránea.

patrocinar

Se generaron 300 tuplas. 200 fueron generadas de manera secuencial y 100 de manera aleatoria y sin duplicados. El atributo nombrePatrocinador fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación patrocinador. El atributo nombreDisciplina fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación disciplina. Relación con llave foráneas.

supervisar

Se generaron 200 tuplas.

El atributo nombrePasaporte fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación juez.

El atributo nombreDisciplina fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación disciplina.

Relación con llave foráneas.

practicar

Se generaron 1000 tuplas.

El atributo nombrePasaporte fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación atleta.

El atributo nombreDisciplina fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación disciplina.

Se agregaron 6 tuplas para satisfacer ciertas consultas.

Resultado final 1006 tuplas.

Relación con llave foráneas.

entrenar

Se generaron 1000 tuplas.

El atributo nombrePasaporteA fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación atleta.

El atributo nombrePasaporteE fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación entrenador.

Relación con llave foráneas.

participar

Se generaron 1200 tuplas.

1000 fueron generadas de manera secuencial y 200 de manera aleatoria.

El atributo nombrePasaporte fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación atleta.

El atributo idEvento fue generado con el tipo de dato *Dataset Column* de manera secuencial en referencia a la relación evento.

Relación con llave foráneas.

7. Consultas

7.1. Consulta 1

La información de todos los atletas que hayan ganado alguna medalla. Así como, un conteo de las medallas de oro, plata y bronce que ganaron. Se muestra la información ordenada con respecto a las medallas:

```

-- La informacion de todos los atletas que hayan ganado alguna medalla. Asi como un conteo de las medallas
-- de oro, plata y bronce que ganaron. La informacion debera ser ordenada con respecto a las medallas, es
-- decir primero oro, despues plata y al final bronce.

select numeropaspporte,nombrepais, fechanacimiento, nacionalidad, nombre, primerapellido, segundoapellido, genero,
       sum(case when tipo = 'Oro' then 1 else 0 end) as oro,
       sum(case when tipo = 'Plata' then 1 else 0 end) as plata,
       sum(case when tipo = 'Bronce' then 1 else 0 end) as bronce
  from (medalla natural join atleta)
 group by numeropaspporte, nombrepais, fechanacimiento, nacionalidad, nombre, primerapellido, segundoapellido, genero
 order by oro desc, plata desc, bronce desc ;

```

numeropaspporte	nombrepais	fechanacimiento	nacionalidad	nombre	primerapellido	segundoapellido	genero	oro	plata	bronce
17M0AA40SU	United States	1988-02-11	Puerto Rico	Alyce	Isakovic	Willment	F	1	1	0
051HEFSZGT	United States	2001-10-20	Eritrea	Trever	Gremi	Ripen	M	1	0	1
1PT6ANLYKD	United States	2002-03-06	United States	Amelita	Mallabon	Barnet	F	1	0	0
1GJW6AN7TG	United States	1985-10-28	Monaco	Rowena	Goldhawk	Bownde	F	1	0	0
03CAULVONP	United States	2002-11-10	South Africa	Haily	Cuthill	Moxley	M	1	0	0
07ZUYYSKRG	Mexico	2002-03-25	United States	Iyell	Oldham	Worcs	M	1	0	0
05DNLLELVY	Mexico	2008-11-29	United States	Jessamyn	Brazenor	Beckwith	F	1	0	0
0R3V8A39T	Mexico	1990-11-16	United States	Nicolas	Jozeftowicz	Kegley	M	1	0	0
14ONB190R6	Mexico	2002-06-06	Belgium	Arny	Chessell	Blunden	M	1	0	0
1NTB1G07S8	Spain	2009-12-26	Japan	Mersey	Enion	Stoddart	F	1	0	0
6CVTEAFFQ2	United States	1985-01-23	Zimbabwe	Stanwood	Savary	Carlton	M	1	0	0
1LFTRGLF7T	United States	1980-07-20	Kuwait	Humphrey	Hurlestone	Pickerill	M	1	0	0
1CDW8DMSCY	Mexico	1982-03-29	Spain	Donna	Netti	Manwell	F	1	0	0
1KOP1BPO1P	Japan	1982-08-23	Mexico	Norry	Woolrich	Joskowitz	M	1	0	0
1NQJDXZDP2	Spain	2002-02-12	Japan	Elyse	Towl	Wakely	F	1	0	0
0NNM4818GRG	Japan	2004-06-26	Guam	Kalvin	Ricket	Whitmore	M	1	0	0
168TPPRLS9	United States	1987-06-18	United States	Sybila	Mowson	Whittlesea	F	1	0	0
0457XEB56A	Japan	2002-07-06	Mexico	Arlen	Baignard	Patesel	F	1	0	0
0Y90HJ0630	United States	1999-02-15	United States	Valma	Hutchison	Arnson	F	1	0	0
01VRSLEUWP	United States	2001-02-24	United States	Chevalier	Rowston	Sales	M	1	0	0
1U4RSS5WHZ	United States	1996-05-30	United States	Aubrette	Furtado	Genike	F	1	0	0
0GN2X4CG3B	United States	1995-09-06	Seychelles	Marietta	Seneschault	Gibanks	F	1	0	0
0C7RZFGET5	Japan	2009-08-15	Ireland	Rikki	Dymott	Skures	F	1	0	0
1QKVVA7OEY	United States	1998-09-13	Spain	Garey	Klosa	Dolton	M	1	0	0
0VAAN7HJU7	Japan	2003-06-10	Mexico	Lancelot	Gladbach	Becconsall	M	1	0	0
15LFDB31L4	Japan	2005-08-05	United States	Demetre	Syrett	Waldrum	M	1	0	0
140EHSJASL	Mexico	2005-08-25	Indonesia	Gurdy	Heinrich	Angus	M	1	0	0
14V2H6P1OD	Mexico	1985-10-02	South Georgia and the South Sandwich Islands	Wendell	Cahalin	Attack	M	1	0	0
21M3MKNWKW	United States	1985-08-19	United States	Ousenia	Schepero	Leaver	F	1	0	0

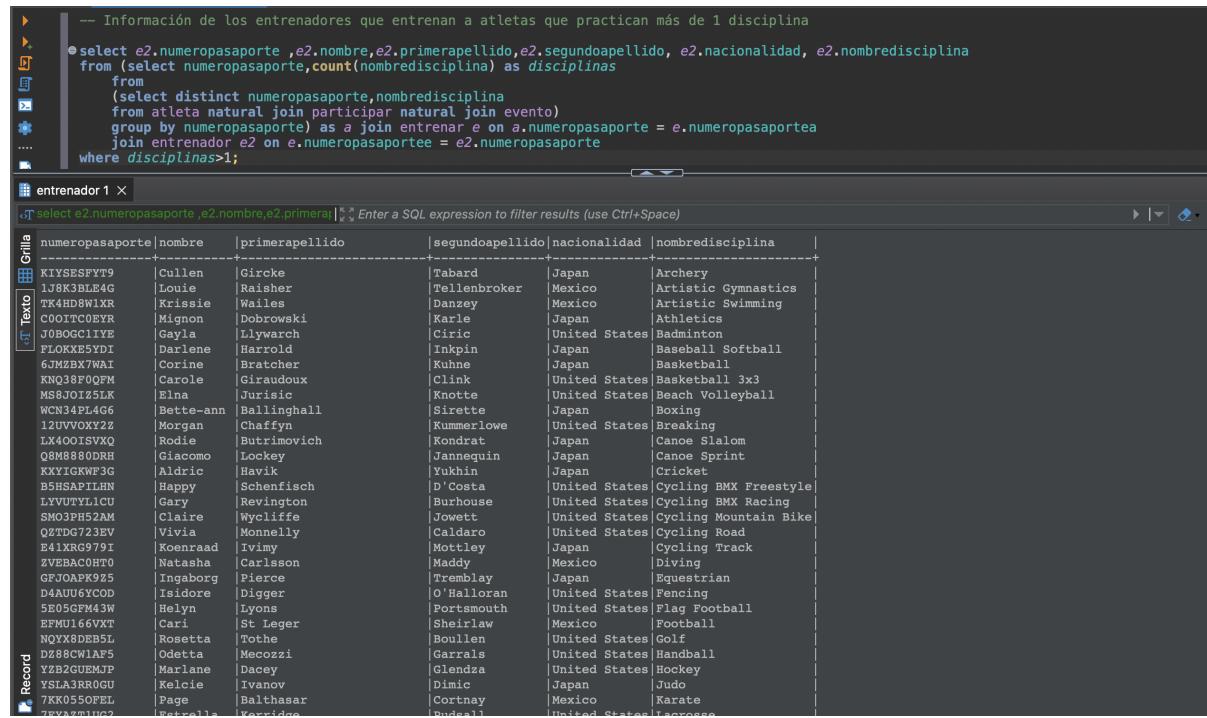
Figura 14: Consulta 1

Dado el resultado de la consulta, se muestra que el máximo ganador, hasta ahora, tiene una medalla de oro y una de plata. Así, se entiende que ningún atleta ha ganado más de una medalla de oro. Además, el máximo ganador hasta ahora es 'Alyce Isakovic Willment'.

Mostrar esta información es importante, pues así se conocen los atletas que más éxito han tenido en los juegos olímpicos. Ya sea para que sus naciones o la misma organización pueda entregar premios o compensaciones a dichos atletas.

7.2. Consulta 2

La información de los entrenados quienes entrena a algún atleta participante en más de una disciplina.



```
-- Información de los entrenadores que entrena a atletas que practican más de 1 disciplina
select e2.numeropasaporte ,e2.nombre,e2.primerapellido,e2.segundoapellido, e2.nacionalidad, e2.nombrerdisciplina
from (select numeropasaporte,count(nombrerdisciplina) as disciplinas
      from
        (select distinct numeropasaporte,nombrerdisciplina
         from atleta natural join participar natural join evento)
       group by numeropasaporte) as a join entrenar e on a.numeropasaporte = e.numeropasaportee
      join entrenador e2 on e.numeropasaportee = e2.numeropasaporte
      where disciplinas>1;
```

The screenshot shows a database query interface with the following details:

- Query Text:**

```
-- Información de los entrenadores que entrena a atletas que practican más de 1 disciplina
select e2.numeropasaporte ,e2.nombre,e2.primerapellido,e2.segundoapellido, e2.nacionalidad, e2.nombrerdisciplina
from (select numeropasaporte,count(nombrerdisciplina) as disciplinas
      from
        (select distinct numeropasaporte,nombrerdisciplina
         from atleta natural join participar natural join evento)
       group by numeropasaporte) as a join entrenar e on a.numeropasaporte = e.numeropasaportee
      join entrenador e2 on e.numeropasaportee = e2.numeropasaporte
      where disciplinas>1;
```
- Results:**

numeropasaporte	nombre	primerapellido	segundoapellido	nacionalidad	nombrerdisciplina
K1Y5E5FT9	Cullen	Gircke	Tabard	Japan	Archery
IJ8K3BLE4G	Louie	Raisher	Tellenbroker	Mexico	Artistic Gymnastics
TX4HD8W1XR	Krissie	Waines	Danzy	Mexico	Artistic Swimming
CO0ITC05YR	Mignon	Dobrowski	Karle	Japan	Athletics
J0BOGCL1YE	Gayla	Llywarch	Ciric	United States	Badminton
FLOKXE5YDI	Darlene	Harrowd	Inkpin	Japan	Baseball Softball
6JM2BX7WAI	Corine	Bratcher	Kuhne	Japan	Basketball
KNQ38F00FM	Carole	Giraudoux	Clink	United States	Basketball 3x3
MS8J01ZSLK	Elna	Jurisic	Knotte	United States	Beach Volleyball
WCN34PL46G	Bette-ann	Ballinghall	Sirette	Japan	Boxing
I2UVVOVX2Z	Morgan	Chaffyn	Kummerlowe	United States	Breaking
LX400ISVXQ	Rodie	Butrimovich	Kondrat	Japan	Canoe Slalom
Q9M8880DRH	Giacomo	Lockey	Janequin	Japan	Canoe Sprint
KXYIIGKWF3G	Aldric	Havik	Yukhin	Japan	Cricket
B5HSAP1LHN	Happy	Schenfisch	D' Costa	United States	Cycling BMX Freestyle
LYVUTYLICU	Gary	Revington	Burhouse	United States	Cycling BMX Racing
SM03PH52AM	Claire	Wycliffe	Jowett	United States	Cycling Mountain Bike
Q2TDG723EV	Vivian	Monnelly	Caldaro	United States	Cycling Road
E41XRG979I	Koenraad	Ivimi	Mottley	Japan	Cycling Track
ZVEBAC0HT0	Natasha	Carlsson	Maddy	Mexico	Diving
GFJOAAPK9Z5	Ingaborg	Pierce	Tremblay	Japan	Equestrian
D4AU6UYCOD	Isidore	Diger	O' Halloran	United States	Fencing
5E05GFM43W	Helyn	Lyons	Portsmouth	United States	Flag Football
EFMU166VXT	Cari	St Leger	Sheirlaw	Mexico	Football
NQYX8DEB5L	Rosetta	Tothe	Boullent	United States	Golf
D288CW1AF5	Odetta	Mecozzi	Garrals	United States	Handball
YZB2GUENJP	Marlane	Dacey	Glendza	United States	Hockey
YSLA3RR0GU	Kelcie	Ivanov	Dimic	Japan	Judo
7KK055OFEL	Page	Balthasar	Cortney	Mexico	Karate
7KXAZM1NC2	Estralla	Kerridge	Dudeall	United States	Lacrosse

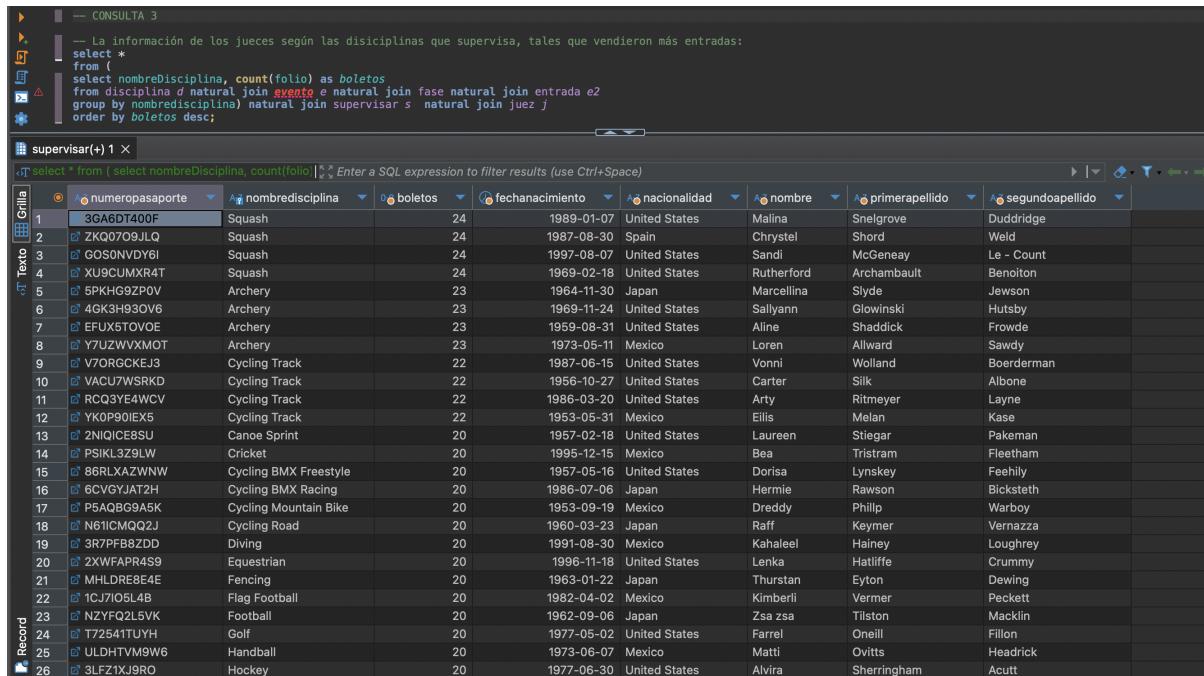
Figura 15: Consulta 2

Dado que un entrenador puede entrena a diversos atletas, entonces se da el caso en que no necesariamente por cada atleta se va a tener el mismo número de entrenadores de acuerdo a la disciplina.

Con esta información se puede empezar a tener una idea de la distribución entre entrenadores y sus atletas. Más aún, ver si los atletas que practican más de una disciplina, tiene más de un entrenador; si dicho entrenador entrena a diversos atletas o es único.

7.3. Consulta 3

La información de los jueces, ordenados según la disciplina , que supervisan, tal que haya vendido más boletos.



```
-- CONSULTA 3
-- La información de los jueces según las disciplinas que supervisa, tales que vendieron más entradas:
select *
from (
select nombreDisciplina, count(folio) as boletos
from disciplina d natural join evento e natural join fase natural join entrada e2
group by nombreDisciplina) natural join supervisar s natural join juez j
order by boletos desc;
```

The screenshot shows a MySQL Workbench interface with a query editor and a results grid. The query retrieves information about judges based on the number of tickets sold, ordered by the number of tickets sold in descending order. The results grid contains columns for: numeropaspasaporte, nombrerdisciplina, boletos, fechanacimiento, nacionalidad, nombre, primerapellido, and segundoapellido. The data includes various sports like Squash, Archery, Cycling, Diving, Equestrian, Fencing, Flag Football, Football, Golf, Handball, and Hockey, along with their respective judges and ticket counts.

	numeropaspasaporte	nombrerdisciplina	boletos	fechanacimiento	nacionalidad	nombre	primerapellido	segundoapellido
1	3GA6DT400F	Squash	24	1989-01-07	United States	Malina	Snelgrove	Duddridge
2	ZKQ0709JLQ	Squash	24	1987-08-30	Spain	Chrystel	Shord	Weld
3	GOS0NVDY6I	Squash	24	1997-08-07	United States	Sandi	McGeneay	Le - Count
4	XU9CUMXR4T	Squash	24	1969-02-18	United States	Rutherford	Archambault	Benilton
5	5PKHG9ZP0V	Archery	23	1964-11-30	Japan	Marcellina	Slyde	Jewson
6	4GK3H930V6	Archery	23	1969-11-24	United States	Sallyann	Glowinski	Hutsby
7	EFUX5TOV0E	Archery	23	1959-08-31	United States	Aline	Shaddick	Frowde
8	Y7UZWVXMOT	Archery	23	1973-05-11	Mexico	Loren	Allward	Sawdy
9	V7ORGCKEJ3	Cycling Track	22	1987-06-15	United States	Vonni	Wolland	Boerderman
10	VACU7WSRKD	Cycling Track	22	1956-10-27	United States	Carter	Silk	Albone
11	RCQ3YE4WCV	Cycling Track	22	1986-03-20	United States	Arty	Ritmeyer	Layne
12	YKOP90IEX5	Cycling Track	22	1953-05-31	Mexico	Ellis	Melan	Kase
13	2NIQICE8SU	Canoe Sprint	20	1957-02-18	United States	Laureen	Stiegar	Pakeman
14	PSIKL3ZBLW	Cricket	20	1995-12-15	Mexico	Bea	Tristram	Fleetham
15	86RLXAZWNW	Cycling BMX Freestyle	20	1957-05-16	United States	Doris	Lynskey	Feehily
16	6CVGYJAT2H	Cycling BMX Racing	20	1986-07-06	Japan	Hermie	Rawson	Bicksteth
17	PSAQBGSA5K	Cycling Mountain Bike	20	1953-09-19	Mexico	Dredy	Phillip	Warboy
18	N61CMQQ2J	Cycling Road	20	1960-03-23	Japan	Raff	Keymer	Vernazza
19	3R7PFB8ZDD	Diving	20	1991-08-30	Mexico	Kahaleel	Hainey	Loughrey
20	2XWFAPR4S9	Equestrian	20	1996-11-18	United States	Lenka	Hattiffe	Crummy
21	MHLDREBE4E	Fencing	20	1963-01-22	Japan	Thurstan	Eyton	Dewing
22	1CJ7I05L4B	Flag Football	20	1982-04-02	Mexico	Kimberli	Vermer	Peckett
23	NZYFQ2L5VK	Football	20	1962-09-06	Japan	Zsa zsa	Tilston	Macklin
24	T72541TUYH	Golf	20	1977-05-02	United States	Farrel	Oneill	Fillon
25	ULDHTVM9W6	Handball	20	1973-06-07	Mexico	Matti	Ovitts	Headrick
26	3LFZ1XJ9RO	Hockey	20	1977-06-30	United States	Alvira	Sherringham	Acutt

Figura 16: Consulta 3

De acuerdo al resultado de la consulta, se observa que la disciplina más vendida por el momento es *Squash* seguida por *Archery*. Posteriormente, se tienen disciplinas con 22 y 20 boletos vendidos cada una.

Además, con dicha información se tiene un enfoque sobre los jueces que han supervisado las disciplinas más vistas. Así, se puede tomar decisiones sobre los mismos o sí se requieren más jueces, pues resulta de interés que dichas disciplinas tengas mejores y más jueces.

7.4. Consulta 4

Información de las localidades que más han sido utilizadas en fase 4 ordenadas por uso y aforo.

```
/* -- Información de las localidades que más han sido utilizadas en fase 4 ordenadas por uso y aforo
select *
from (
    select idLocalidad, count(*) as numeroUsos
    from evento
    natural join (
        select idLocalidad, idEvento
        from fase
        natural join evento
        where nombreFase = 'Fase 4')
    group by idLocalidad)
natural join localidad
order by numeroUsos desc, aforo desc;
```

eventos(+) 1 X

	idLocalidad	numeroUsos	nombrepais	calle	numero	ciudad	nombre	tipo	aforo
1	17	3	United States	Avalon Blvd	18,400	Carson	Dignity Health Sports Park	Parque	30,000
2	14	3	United States	Burbank Blvd	17,017	Encino	Sepulveda Basin Park	Parque	15,000
3	16	3	United States	Avalon Blvd	18,400	Carson	Dignity Health Sports Park	Parque	15,000
4	19	3	United States	Ocean Blvd	300	Long Beach	Long Beach Arena	Arena	12,000
5	15	3	United States	Avalon Blvd	18,400	Carson	Dignity Health Sports Park	Parque	10,000
6	13	3	United States	Burbank Blvd	17,017	Encino	Sepulveda Basin Park	Parque	8,000
7	18	3	United States	Avalon Blvd	18,400	Carson	Velo Sports Center	Velodromo	6,000
8	20	3	United States	PI	35	Long Beach	Belmont Veterans Memorial Pier	Muelle	6,000
9	11	3	United States	Grand Ave	200	Los Angeles	Grand Park	Parque	5,000
10	12	3	United States	Burbank Blvd	17,017	Encino	Sepulveda Basin Park	Parque	3,000
11	1	2	United States	S Figueroa St	3,911	Los Angeles	Los Angeles Memorial Coliseum	Estadio	70,000
12	2	2	United States	S Figueroa St	3,939	Los Angeles	BMO Stadium	Estadio	22,000
13	3	2	United States	Childs Way	1,021	Los Angeles	Dedeaux Field	Estadio	20,000
14	9	2	United States	S Figueroa St	1,111	Los Angeles	Crypto Arena	Recinto	18,000
15	4	2	United States	S Figueroa St	3,400	Los Angeles	Galean Center	Estadio Cubierto	10,300
16	8	2	United States	S Figueroa St	1,201	Los Angeles	Los Angeles Convention Center	Centro de Convenciones	9,000
17	7	2	United States	S Figueroa St	1,201	Los Angeles	Los Angeles Convention Center	Centro de Convenciones	8,000
18	6	2	United States	S Figueroa St	1,201	Los Angeles	Los Angeles Convention Center	Centro de Convenciones	7,000
19	10	2	United States	Chick Hearn	777	Los Angeles	Peacock Theater	Auditorio	7,000
20	140	2	United States	Johnson	3	Urbub	optimal	infrastructure	6,278
21	5	2	United States	S Figueroa St	1,201	Los Angeles	Los Angeles Convention Center	Centro de Convenciones	5,000
22	118	2	United States	Corry	6,435	Haenam	customer loyalty	maximized	2,782
23	135	1	Netherlands	Brickson Park	673	Hanyin Chenggu	didactic	actuating	8,561
24	132	1	Angola	Monica	42	Padre Paraiso	responsive	flexibility	8,282
25	137	1	Seychelles	Northland	975	Phra Yuen	internet solution	Compatible	6,654
26	136	1	Denmark	Cascade	23	Slobodka	framework	radical	5,591
27	146	1	Luxembourg	Di Loreto	31	Edéa	Exclusive	Versatile	3,186
28	134	1	Italy	American	18,048	Manukaka	motivating	content-based	1,399
29	133	1	Indonesia	Sauthoff	49,777	Morinville	Ergonomic	Synergized	778
30	444	1	United States	S Figueroa St	391	Los Angeles	Los Sant	centro	4

Figura 17: Consulta 4

De acuerdo al resultado de la consulta, se puede observar que el mayor número de eventos de fase 4 realizado es de 3, en donde hay 10 localidades que comparten dicho número de usos. Dentro de estas, podemos destacar que ninguna es de tipo estadio, y que el aforo de estas no supera los 30,000 espectadores. Pues la localidad con mayor número de usos y mayor aforo es la localidad con idLocalidad: 17, de nombre Dignity Health Sports Park.

7.5. Consulta 5

Disciplinas con más patrocinadores que tienen el mayor número de espectadores presenciales.

The screenshot shows a database interface with two tables of results. The top table, titled 'disciplina 1 x', lists disciplines with their number of sponsors and spectators. The bottom table, titled 'Record', lists disciplines with their number of sponsors and spectators, ordered by the number of spectators in descending order. Both tables have columns for 'nombredisciplina', 'numeropatrocinadores', and 'numeroespectadores'.

	nombredisciplina	numeropatrocinadores	numeroespectadores
1	Karate	11	20
2	Diving	9	20
3	Baseball Softball	8	20
4	Handball	8	20
5	Cycling Mountain Bike	8	20
6	Taekwondo	8	20
7	Archery	7	23
8	Canoe Slalom	7	20
9	Rugby Sevens	7	20
10	Tennis	7	20
11	Artistic Gymnastics	7	20
12	Judo	7	20
13	Hockey	7	20
14	Table Tennis	7	20
15	Water Polo	7	20
16	Sport Climbing	7	20
17	Marathon Swimming	7	20

	nombredisciplina	numeropatrocinadores	numeroespectadores
21	Skateboarding	6	20
22	Badminton	6	20
23	Beach Volleyball	6	20
24	Cycling BMX Freestyle	6	20
25	Cycling BMX Racing	6	20
26	Football	6	20
27	Lacrosse	6	20
28	Shooting	6	20
29	Fencing	5	20
30	Artistic Swimming	5	20
31	Cycling Road	5	20
32	Modern Pentathlon	5	20
33	Rhythmic Gymnastics	5	20
34	Rowing	5	20
35	Basketball 3x3	5	20
36	Basketball	5	20
37	Athletics	5	20
38	Volleyball	5	20
39	Golf	5	20
40	Equestrian	5	20
41	Sailing	5	20
42	Breaking	5	20
43	Boxing	5	20
44	Surfing	5	20
45	Cricket	4	20
46	Trampoline	4	20
47	Weightlifting	4	20
48	Canoe Sprint	4	20
49	Swimming	4	20
50	Triathlon	4	20

Figura 18: Consulta 5

Esta consulta es de gran utilidad para poder definir que disciplinas son las que son más deseadas por los patrocinadores, además, con esta es posible determinar algunas disciplinas que tienen potencial para los distintos patrocinadores, pues serían aquellas que posean un número reducido de patrocinadores pero tengan un mayor número de espectadores.

Analizando la consulta, podemos observar que el número de espectadores para todas las disciplinas es de 20, sin embargo, el número de patrocinadores es distinto. Podemos observar que Karate es la disciplina con mayor número de patrocinadores, mientras que las disciplinas con menor número de patrocinadores son Weightlifting, CanoeSprint, Swimming y Triathlon. Por lo que estas últimas podrían ser de interés para los patrocinadores.

7.6. Consulta 6

Eventos que se realizan fuera de Los Angeles pero dentro de Estados Unidos

```
-- Eventos que se realizan fuera de Los Angeles pero dentro de Estados Unidos
select idEvento, nombreDisciplina, nombre, ciudad, fecha
from localidad natural join evento
where nombrePais = 'United States' and ciudad <> 'Los Angeles'
```

idevento	nombredisciplina	nombre	ciudad	fecha
12	Canoe Slalom	Sepulveda Basin Park	Encino	2028-07-21
13	Canoe Sprint	Sepulveda Basin Park	Encino	2028-07-27
14	Cricket	Sepulveda Basin Park	Encino	2028-07-28
15	Cycling BMX Freestyle	Dignity Health Sports Park	Carson	2028-07-19
16	Cycling BMX Racing	Dignity Health Sports Park	Carson	2028-07-25
17	Cycling Mountain Bike	Dignity Health Sports Park	Carson	2028-07-14
18	Cycling Road	Velo Sports Center	Carson	2028-07-27
19	Cycling Track	Long Beach Arena	Long Beach	2028-07-16
20	Diving	Belmont Veterans Memorial Pier	Long Beach	2028-07-28
32	Modern Pentathlon	Sepulveda Basin Park	Encino	2028-07-29
33	Rhythmic Gymnastics	Sepulveda Basin Park	Encino	2028-07-26
34	Rowing	Sepulveda Basin Park	Encino	2028-07-29
35	Rugby Sevens	Dignity Health Sports Park	Carson	2028-07-22
36	Sailing	Dignity Health Sports Park	Carson	2028-07-20
37	Shooting	Dignity Health Sports Park	Carson	2028-07-27
38	Skateboarding	Velo Sports Center	Carson	2028-07-24
39	Sport Climbing	Long Beach Arena	Long Beach	2028-07-20
40	Squash	Belmont Veterans Memorial Pier	Long Beach	2028-07-19
52	Artistic Gymnastics	Sepulveda Basin Park	Encino	2028-07-29
53	Artistic Swimming	Sepulveda Basin Park	Encino	2028-07-25
54	Athletics	Sepulveda Basin Park	Encino	2028-07-18
55	Badminton	Dignity Health Sports Park	Carson	2028-07-19
56	Baseball Softball	Dignity Health Sports Park	Carson	2028-07-26
57	Basketball	Dignity Health Sports Park	Carson	2028-07-27
58	Basketball 3x3	Velo Sports Center	Carson	2028-07-27
59	Beach Volleyball	Long Beach Arena	Long Beach	2028-07-19
60	Boxing	Belmont Veterans Memorial Pier	Long Beach	2028-07-28
72	Fencing	Sepulveda Basin Park	Encino	2028-07-24
73	Flag Football	Sepulveda Basin Park	Encino	2028-07-26
74	Football	Sepulveda Basin Park	Encino	2028-07-22
75	Golf	Dignity Health Sports Park	Carson	2028-07-19
155	Badminton	Dignity Health Sports Park	Carson	2028-07-19
156	Baseball Softball	Dignity Health Sports Park	Carson	2028-07-19
157	Basketball	Dignity Health Sports Park	Carson	2028-07-15
158	Basketball 3x3	Velo Sports Center	Carson	2028-07-22
159	Beach Volleyball	Long Beach Arena	Long Beach	2028-07-21
160	Boxing	Belmont Veterans Memorial Pier	Long Beach	2028-07-18
172	Fencing	Sepulveda Basin Park	Encino	2028-07-22
173	Flag Football	Sepulveda Basin Park	Encino	2028-07-22
174	Football	Sepulveda Basin Park	Encino	2028-07-25
175	Golf	Dignity Health Sports Park	Carson	2028-07-27
176	Handball	Dignity Health Sports Park	Carson	2028-07-24
177	Hockey	Dignity Health Sports Park	Carson	2028-07-26
178	Judo	Velo Sports Center	Carson	2028-07-18
179	Karate	Long Beach Arena	Long Beach	2028-07-18
180	Lacrosse	Belmont Veterans Memorial Pier	Long Beach	2028-07-20
192	Swimming	Sepulveda Basin Park	Encino	2028-07-15
193	Table Tennis	Sepulveda Basin Park	Encino	2028-07-28
194	Taekwondo	Sepulveda Basin Park	Encino	2028-07-15
195	Tennis	Dignity Health Sports Park	Carson	2028-07-20
196	Trampoline	Dignity Health Sports Park	Carson	2028-07-28
197	Triathlon	Dignity Health Sports Park	Carson	2028-07-28
198	Volleyball	Velo Sports Center	Carson	2028-07-15
199	Water Polo	Long Beach Arena	Long Beach	2028-07-19
200	Weightlifting	Belmont Veterans Memorial Pier	Long Beach	2028-07-24
201	Archery	customer loyalty	Haenam	2028-07-24
202	Swimming	customer loyalty	Haenam	2028-07-23
203	Swimming	optimal	Urbub	2028-07-21
204	Swimming	optimal	Urbub	2028-07-22

Figura 19: Consulta 6

La consulta nos regresa 90 filas que corresponden a los distintos eventos que se realizan fuera de Los Angeles, podemos hacer notar que dichos eventos requieren de instalaciones especiales o de sedes específicas. Por lo que es útil poder obtener dichos eventos junto con sus ubicaciones y fechas para poder informar a los participantes y espectadores las ubicaciones de dichos eventos y poder planificar de mejor manera. Basándonos en los resultados, todos los eventos externos a Los Angeles se realizarán en las ciudades de Carson, Long Beach, Encinol, Haenam y Urbub.

7.7. Consulta 7

La información de los eventos que se llevan a cabo en la localidad con mayor cantidad de eventos, ordenados de acuerdo a su fecha de inicio

```
-- La información de los eventos que se llevan a cabo en la localidad con mayor cantidad de eventos, ordenados de acuerdo a su fecha de inicio
select idEvento, evento.idLocalidad, nombreDisciplina, fecha, horaInicio from (
    select max(num_eventos)
        from (select idLocalidad, count(idEvento) as num_eventos
            from evento
            group by idLocalidad)) as m join (select idLocalidad, count(idEvento) as num_eventos
        from evento
        group by idLocalidad) as a on m.max = a.num_eventos join evento on a.idLocalidad = evento.idlocalidad
order by fecha asc;
```

Grid	idEvento	idLocalidad	nombreDisciplina	fecha	horaInicio
1		17	Cycling Mountain Bike	2028-07-14	14:10:00
2		43	Table Tennis	2028-07-14	15:23:00
3		8	Basketball 3x3	2028-07-14	19:29:00
4		64	Cricket	2028-07-14	09:30:00
5		83	Rhythmic Gymnastics	2028-07-14	11:40:00
6		22	Fencing	2028-07-14	07:09:00
7		23	Flag Football	2028-07-14	05:49:00
8		121	Equestrian	2028-07-14	16:57:00
9		147	Triathlon	2028-07-14	09:37:00
10		128	Judo	2028-07-14	09:37:00
11		137	Shooting	2028-07-14	20:42:00
12		136	Sailing	2028-07-14	23:30:00
13		132	Modern Pentathlon	2028-07-14	13:45:00
14		98	Volleyball	2028-07-14	15:20:00
15		151	Archery	2028-07-14	21:23:00
16		78	Judo	2028-07-14	22:54:00
17		157	Basketball	2028-07-15	12:03:00
18		148	Volleyball	2028-07-15	05:08:00
19		194	Taekwondo	2028-07-15	08:50:00
20		192	Swimming	2028-07-15	17:06:00
21		127	Hockey	2028-07-15	15:15:00
22		140	Squash	2028-07-15	05:30:00
23		107	Basketball	2028-07-15	13:17:00
24		100	Weightlifting	2028-07-15	06:46:00
25		28	Judo	2028-07-15	11:11:00
26		109	Beach Volleyball	2028-07-15	10:09:00
27		123	Flag Football	2028-07-15	11:15:00
28		198	Volleyball	2028-07-15	06:29:00
29		81	Marathon Swimming	2028-07-15	17:26:00
30		169	Cycling Track	2028-07-16	06:17:00
31		19	Cycling Track	2028-07-16	07:18:00
32		79	Karate	2028-07-16	11:58:00

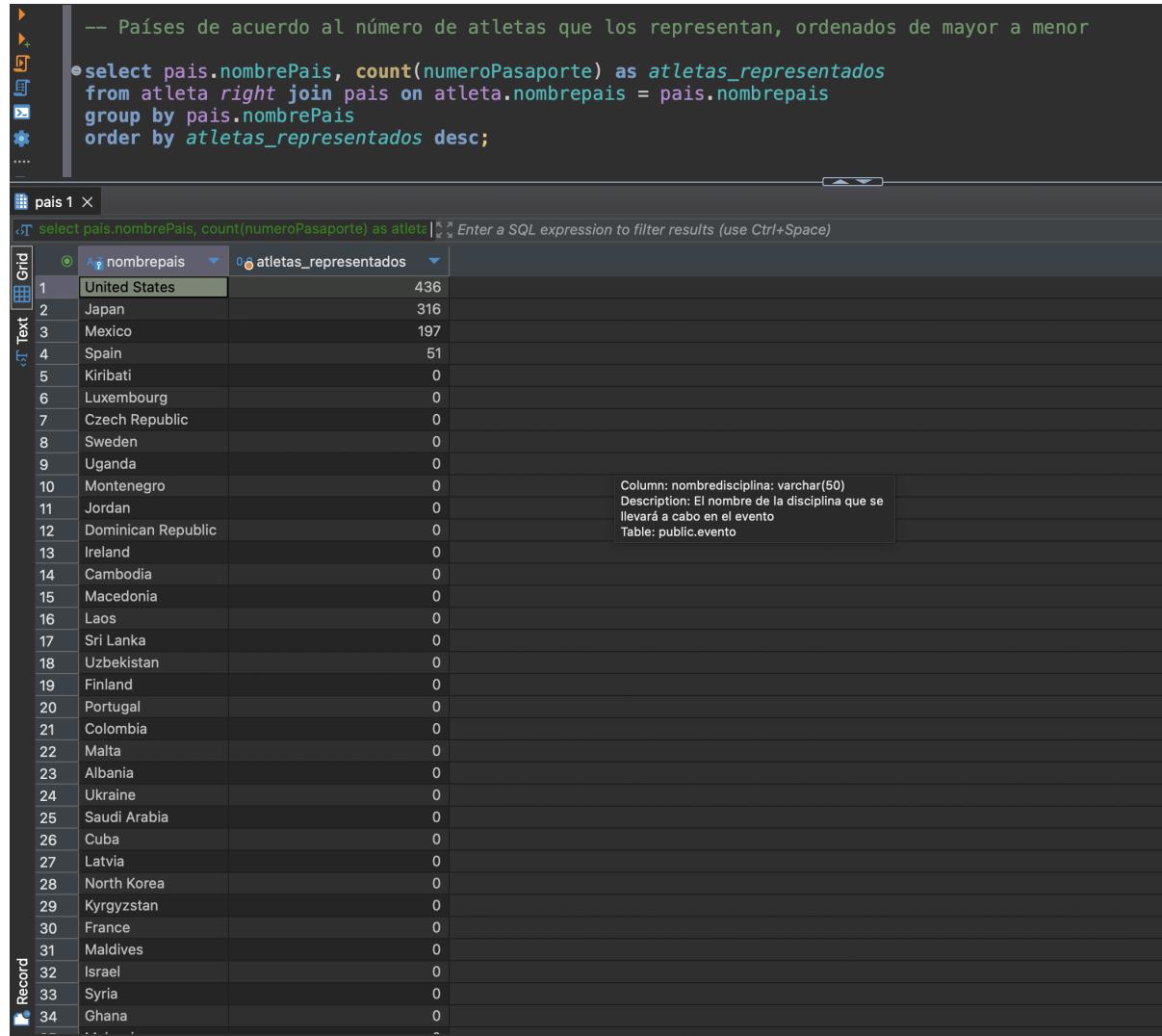
Figura 20: Consulta 7

Esta información es de utilidad para conocer hasta cuantos eventos puede albergar una localidad, y en este caso, la información de estos eventos, es decir, su fecha, el tipo, etc. Esto para poder planificar de una manera más eficiente alguna edición futura de los juegos olímpicos en ese mismo lugar.

Dado que nuestra base de datos, el máximo de eventos en una localidad es compartido por varias localidades, el resultado de esta consulta nos da la información de todos los eventos que ocurren en todas estas localidades.

7.8. Consulta 8

Países de acuerdo al número de atletas que los representan, ordenados de mayor a menor



```
-- Países de acuerdo al número de atletas que los representan, ordenados de mayor a menor
select pais.nombrepais, count(numeroPasaporte) as atletasRepresentados
from atleta right join pais on atleta.nombrepais = pais.nombrepais
group by pais.nombrepais
order by atletasRepresentados desc;
```

	nombrepais	atletasRepresentados
1	United States	436
2	Japan	316
3	Mexico	197
4	Spain	51
5	Kiribati	0
6	Luxembourg	0
7	Czech Republic	0
8	Sweden	0
9	Uganda	0
10	Montenegro	0
11	Jordan	0
12	Dominican Republic	0
13	Ireland	0
14	Cambodia	0
15	Macedonia	0
16	Laos	0
17	Sri Lanka	0
18	Uzbekistan	0
19	Finland	0
20	Portugal	0
21	Colombia	0
22	Malta	0
23	Albania	0
24	Ukraine	0
25	Saudi Arabia	0
26	Cuba	0
27	Latvia	0
28	North Korea	0
29	Kyrgyzstan	0
30	France	0
31	Maldives	0
32	Israel	0
33	Syria	0
34	Ghana	0

Figura 21: Consulta 8

Esta consulta sirve para realizar cálculos estadísticos acerca de que países son los que más atletas suelen producir o apoyar, dicha información puede usarse para poder regular la cantidad de atletas por país, que los gobiernos puedan saber las estadísticas de su país y así poder tomar respecto a eso, entre otros posibles usos.

Dado que nuestra base de datos únicamente tiene atletas que representan a Estados Unidos, México y Japón, esos serán los únicos países que aparecen con más de 0, teniendo los demás 0 representantes.

7.9. Consulta 9

Cada disciplina practicada y el género predominante en ella, ordenados por el nombre de la disciplina

```

@with numeroGeneros as (
    select nombreDisciplina, genero, count(genero) as numeroGen
    from practicar natural join atleta
    group by nombreDisciplina, genero),
predominante as (
    select nombreDisciplina, max(numeroGen) as maxG
    from numeroGeneros
    group by nombreDisciplina
    order by nombreDisciplina
)
select predominante.nombreDisciplina, genero as genero_pronominante
from predominante join numeroGeneros on numeroGen = maxG and predominante.nombreDisciplina = numeroGeneros.nombreDisciplina
order by predominante.nombreDisciplina;

```

	A-Z nombreDisciplina	A-Z genero_pronominante
1	Archery	M
2	Artistic Gymnastics	F
3	Artistic Swimming	F
4	Athletics	M
5	Badminton	F
6	Baseball Softball	F
7	Baseball Softball	M
8	Basketball	M
9	Basketball 3x3	F
10	Beach Volleyball	F
11	Boxing	M
12	Breaking	M
13	Canoe Slalom	M
14	Canoe Sprint	F
15	Canoe Sprint	M
16	Cricket	M
17	Cycling BMX Freestyle	M
18	Cycling BMX Racing	F
19	Cycling Mountain Bike	M
20	Cycling Road	M
21	Cycling Road	F
22	Cycling Track	M
23	Diving	F
24	Equestrian	M
25	Fencing	M
26	Flag Football	M
27	Flag Football	F
28	Football	F
29	Golf	F
30	Handball	F
31	Hockey	M

Figura 22: Consulta 9

Dicha consulta puede ser de gran utilidad para poder planificar mejor ciertos eventos y poder realizar varios estudios de acuerdo al resultado que arroje cierta disciplina, con el cuál por ejemplo se puede incentivar una disciplina más hacia cierta demografía para que aumente el número de atletas de cierto género en dicha disciplina en próximas ediciones.

Por nuestra implementación, cuando existe el mismo número de atletas mujeres y hombres en una disciplina, ambas filas se agregan a la tabla, es decir, que si una disciplina x, tiene 10 mujeres y 10 hombres practicantes, entonces en la tabla resultante de la consulta aparecerán (x, M) y (x, F).

7.10. Consulta 10

El atleta más joven de cada disciplina, ordenados de forma ascendente por el nombre de la disciplina.

```

•with fechaMenor as (
    select nombreDisciplina, max(fechaNacimiento) as fechaNacimiento
    from practicar natural join atleta
    group by nombreDisciplina
)
-- Query
select nombreDisciplina, nombre, primerApellido, segundoApellido, fechaNacimiento
from atleta natural join fechaMenor
order by nombreDisciplina ;

```

practicar(+) 1 (10) × evento(+) 1 (11) × fase(+) 1 (12)

select numeropasaporte,nombrepais,fechanacimiento,naciosa | Data filter is not supported

	nombredisciplina	nombre	primerapellido	segundoapellido	fechanacimiento
1	Archery	Alfred	Winspurr	Brilon	2009-11-17
2	Artistic Gymnastics	Elton	Fry	Briskey	2008-08-24
3	Artistic Swimming	Shani	Jenkinson	Thoumas	2009-06-14
4	Artistic Swimming	Donnie	Pensom	Tixall	2009-06-14
5	Athletics	Glyn	Bootland	Kaygill	2009-12-15
6	Badminton	Dolly	Kraft	O'Hederscoll	2009-05-13
7	Baseball Softball	Shani	Jenkinson	Thoumas	2009-06-14
8	Baseball Softball	Donnie	Pensom	Tixall	2009-06-14
9	Basketball	Priscella	Skitterel	Jovey	2009-08-10
10	Basketball 3x3	Layne	Gero	Burdas	2009-11-18
11	Beach Volleyball	Iormina	Lecordier	Baylie	2009-08-31
12	Boxing	Corabelle	Maplethorp	Rawes	2008-04-17
13	Breaking	Vida	Rings	Catlow	2004-07-11
14	Canoe Slalom	Charyl	Hamflett	Mendonca	2008-08-31
15	Canoe Sprint	Rock	Dhenin	Casswell	2008-10-06
16	Cricket	Kliment	Coolson	Surette	2005-07-04
17	Cycling BMX Freestyle	Margarethe	Eager	Morillas	2008-06-11
18	Cycling BMX Freestyle	Salvador	Riditch	Skepper	2008-06-11
19	Cycling BMX Racing	Sarina	Grishinov	Persent	2009-05-17
20	Cycling Mountain Bike	Asia	Dyos	Hullins	2009-12-25
21	Cycling Road	Nannie	Flohard	Gullyes	2009-06-09
22	Cycling Track	Merrill	Danilovich	Gerbl	2009-06-29
23	Diving	Gerik	Kreutzer	Ludgrove	2009-09-06
24	Equestrian	Towny	Brigginshaw	MacKellen	2007-07-22
25	Fencing	Alexei	Tremellier	Angerstein	2007-10-14
26	Flag Football	Andonis	Fishleigh	Franiak	2006-11-13

Figura 23: Consulta 10

Podemos ver que todos los participantes más jóvenes nacieron en los 2000's y algunos son menores de edad. Asimismo, algunos dentro de una disciplina comparten cumpleaños, como en *Cycling BMX Freestyle*.

Esta información puede ser muy útil para saber qué disciplinas tienen menores de edad, lo que podrían cambiar aspectos de la organización para ajustarse a ellos. Además, los atletas más jóvenes suelen recibir más atención por parte del público, por lo que puede ser conveniente conocerlos.

7.11. Consulta 11

Fecha, hora y localidad de la fase 4 de cada disciplina, ordenados de forma ascendente por el nombre de la disciplina y cronología.

	disciplina	localidad	fecha	hora
1	Archery	Grand Park	2028-07-14	21:23:00
2	Archery	Los Sant	2028-07-20	11:18:00
3	Archery	Exclusive	2028-07-22	13:00:00
4	Archery	customer loyalty	2028-07-24	13:00:00
5	Artistic Gymnastics	Sepulveda Basin Park	2028-07-20	08:08:00
6	Artistic Swimming	Sepulveda Basin Park	2028-07-29	12:06:00
7	Athletics	framework	2028-07-24	13:00:00
8	Athletics	Sepulveda Basin Park	2028-07-29	06:38:00
9	Badminton	Dignity Health Sports Park	2028-07-19	23:37:00
10	Baseball Softball	Dignity Health Sports Park	2028-07-19	16:53:00
11	Basketball	Dignity Health Sports Park	2028-07-15	12:03:00
12	Basketball 3x3	Velo Sports Center	2028-07-22	16:55:00
13	Beach Volleyball	Long Beach Arena	2028-07-21	05:06:00
14	Boxing	Belmont Veterans Memorial Pier	2028-07-18	09:59:00
15	Boxing	internet solution	2028-07-23	13:00:00
16	Breaking	Los Angeles Memorial Coliseum	2028-07-22	15:08:00
17	Breaking	motivating	2028-07-24	13:00:00
18	Canoe Slalom	BMO Stadium	2028-07-21	12:46:00
19	Canoe Sprint	Dedeaux Field	2028-07-17	18:12:00
20	Cricket	didactic	2028-07-25	13:00:00
21	Cricket	Galean Center	2028-07-29	14:41:00
22	Cycling BMX Freestyle	Los Angeles Convention Center	2028-07-19	08:09:00
23	Cycling BMX Racing	Los Angeles Convention Center	2028-07-26	19:48:00
24	Cycling Mountain Bike	Los Angeles Convention Center	2028-07-17	09:05:00
25	Cycling Road	Los Angeles Convention Center	2028-07-16	11:09:00
26	Cycling Track	Crypto Arena	2028-07-16	06:17:00
27	Diving	Peacock Theater	2028-07-29	15:33:00
28	Equestrian	Grand Park	2028-07-18	11:41:00
29	Fencing	Sepulveda Basin Park	2028-07-22	23:53:00
30	Flag Football	Sepulveda Basin Park	2028-07-22	16:19:00
31	Football	Sepulveda Basin Park	2028-07-25	07:09:00

Figura 24: Consulta 11

Podemos observar que deportes como *Archery* tienen más de un evento para la fase 4. Asimismo, podemos ver que no todas las disciplinas llegan hasta la fase 4.

Esta consulta nos permite saber los detalles más importantes de los eventos. Esta fase es llamada bastante la atención del público, ya que es muy avanzada.

7.12. Consulta 12

El evento más largo de cada disciplina, ordenados por la duración máxima, el nombre de la disciplina y por cronología.

```

with eventoMaximo as (
    select nombreDisciplina, max(duracionMaxima) as duracionMaxima
    from evento
    group by nombreDisciplina
)
-- Query
select nombreFase, idEvento, nombreDisciplina, duracionMaxima, fecha
from fase natural join evento natural join eventoMaximo
order by duracionMaxima desc, nombreDisciplina, fase, fecha;

```

The screenshot shows a database interface with a SQL query editor at the top and a results table below. The table has the following data:

	nombreFase	idEvento	nombreDisciplina	duracionMaxima	fecha
1	Fase 3	102	Artistic Gymnastics	5	2028-07-23
2	Fase 3	103	Artistic Swimming	5	2028-07-23
3	Fase 3	104	Athletics	5	2028-07-20
4	Fase 3	108	Basketball 3x3	5	2028-07-23
5	Fase 4	158	Basketball 3x3	5	2028-07-22
6	Fase 1	11	Breaking	5	2028-07-18
7	Fase 1	14	Cricket	5	2028-07-28
8	Fase 1	15	Cycling BMX Freestyle	5	2028-07-19
9	Fase 4	166	Cycling BMX Racing	5	2028-07-26
10	Fase 1	18	Cycling Road	5	2028-07-27
11	Fase 2	71	Equestrian	5	2028-07-29
12	Fase 3	121	Equestrian	5	2028-07-14
13	Fase 1	22	Fencing	5	2028-07-14
14	Fase 3	123	Flag Football	5	2028-07-15
15	Fase 2	74	Football	5	2028-07-22
16	Fase 1	26	Handball	5	2028-07-23
17	Fase 3	126	Handball	5	2028-07-19
18	Fase 4	176	Handball	5	2028-07-24
19	Fase 4	178	Judo	5	2028-07-18
20	Fase 4	181	Marathon Swimming	5	2028-07-27
21	Fase 2	88	Skateboarding	5	2028-07-24
22	Fase 3	138	Skateboarding	5	2028-07-27
23	Fase 4	188	Skateboarding	5	2028-07-27
24	Fase 3	140	Squash	5	2028-07-15
25	Fase 3	141	Surfing	5	2028-07-26
26	Fase 2	92	Swimming	5	2028-07-27
27	Fase 4	193	Table Tennis	5	2028-07-28
28	Fase 2	95	Tennis	5	2028-07-29
29	Fase 3	145	Tennis	5	2028-07-25
30	Fase 3	146	Trampoline	5	2028-07-19
31	Fase 1	47	Triathlon	5	2028-07-22
32	Fase 2	97	Triathlon	5	2028-07-19
33	Fase 2	98	Volleyball	5	2028-07-14
34	Fase 4	198	Volleyball	5	2028-07-15
35	Fase 2	99	Water Polo	5	2028-07-25
36	Fase 3	149	Water Polo	5	2028-07-21
37	Fase 2	100	Weightlifting	5	2028-07-15
38	Fase 1	1	Archery	4	2028-07-20
39	Fase 2	51	Archery	4	2028-07-18
40	Fase 4	201	Archery	4	2028-07-24
41	Fase 4	444	Archery	4	2028-07-20
42	Fase 4	155	Badminton	4	2028-07-19
43	Fase 1	7	Basketball	4	2028-07-27
44	Fase 2	57	Basketball	4	2028-07-27

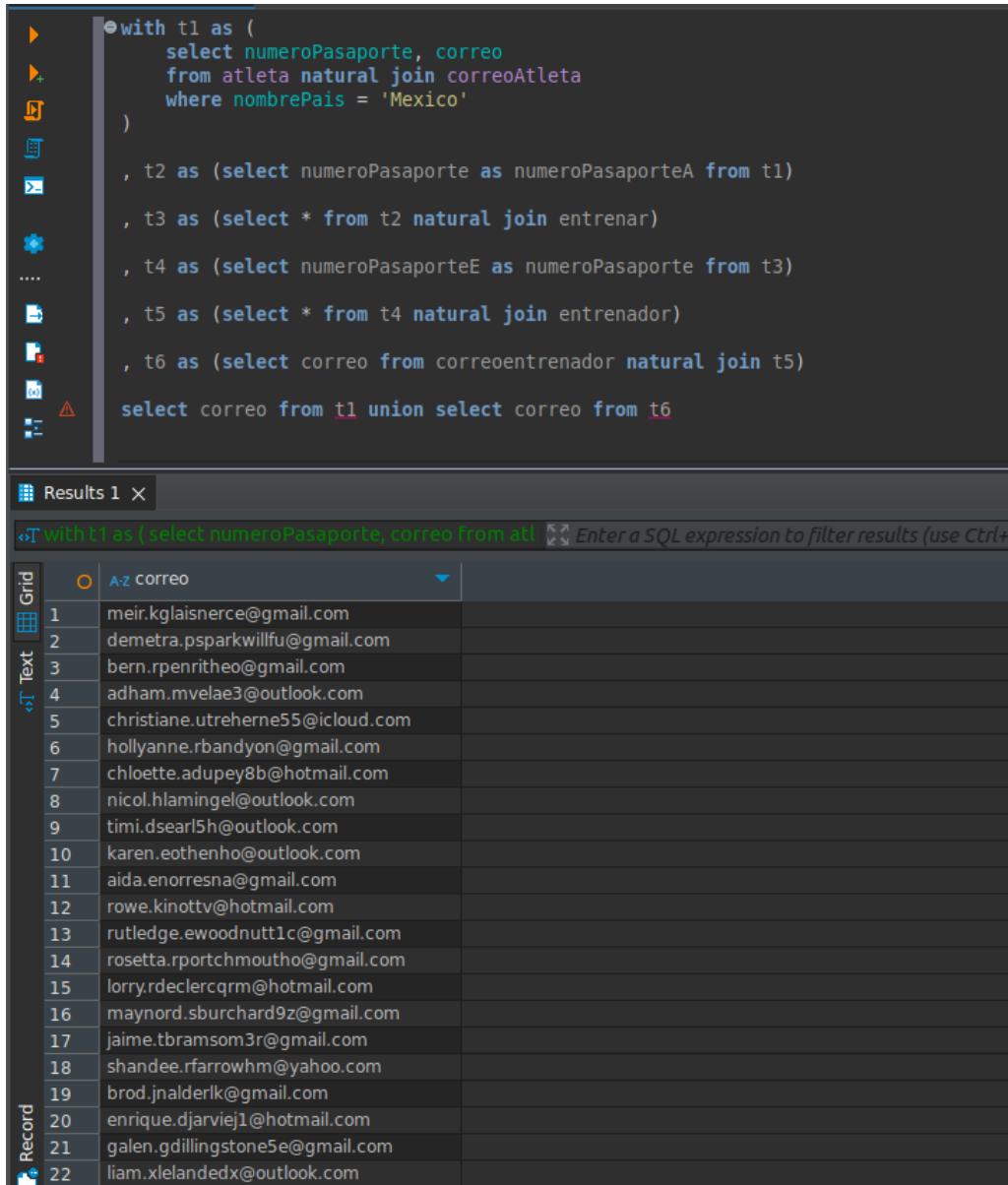
Figura 25: Consulta 12

Notemos que varios deportes tienen fases con la misma duración máxima, siendo de 5 horas la mayor. Notemos que hay disciplinas como *Handball* que tienen más de una fase con la duración mayor. En casos especiales como con *Archery*, se repiten las fases ya que están en eventos distintos.

Con la duración máxima de los eventos de un deporte, podremos conocer las preparaciones necesarias para establecer los tiempos de descanso necesarios y asegurar la comodidad de los asistentes. Además identificamos en la columna *nombreFase* las fases en las que hay que hacer dichas preparaciones, que se realizarán en el evento de la columna *idEvento*.

7.13. Consulta 13

La lista de correos de atletas que representan a México y de los entrenadores que los entrena.



```

with t1 as (
    select numeroPasaporte, correo
    from atleta natural join correoAtleta
    where nombrePais = 'Mexico'
)
, t2 as (select numeroPasaporte as numeroPasaporteA from t1)
, t3 as (select * from t2 natural join entrenar)
, t4 as (select numeroPasaporteE as numeroPasaporte from t3)
, t5 as (select * from t4 natural join entrenador)
, t6 as (select correo from correoentrenador natural join t5)

select correo from t1 union select correo from t6

```

The screenshot shows a SQL query editor interface. At the top, there is a code editor window containing the provided SQL query. Below it is a results grid titled "Results 1 X". The grid has two columns: "Record" (numbered 1 to 22) and "correo" (email address). The "correo" column displays 22 distinct email addresses, all ending in @gmail.com or @outlook.com.

Record	correo
1	meir.kglaisnerce@gmail.com
2	demetra.psparkwillfu@gmail.com
3	bern.rpenritheo@gmail.com
4	adham.mvelae3@outlook.com
5	christiane.utreherne55@icloud.com
6	hollyanne.rbandyon@gmail.com
7	chloette.adupey8b@hotmail.com
8	nicol.hlamingel@outlook.com
9	timi.dsearl5h@outlook.com
10	karen.eotheno@outlook.com
11	aida.enorresna@gmail.com
12	rowe.kinottv@hotmail.com
13	rutledge.ewoodnutt1c@gmail.com
14	rosetta.rportchmouho@gmail.com
15	lorry.rdeclercqrm@hotmail.com
16	maynord.sburchard9z@gmail.com
17	jaime.tbramsom3r@gmail.com
18	shandee.rfarrowhm@yahoo.com
19	brod.jnalderlk@gmail.com
20	enrique.djarviej1@hotmail.com
21	galen.gdillingstone5e@gmail.com
22	liam.xlelandedx@outlook.com

Figura 26: Consulta 13

Esta consulta puede ser fácilmente modificada para obtener la misma información de cualquier otro país. Nos interesa conocer todos los miembros relacionados al país que representan ya que de esta forma como organizadores del evento es más fácil enviar notificaciones o instrucciones específicas que deberán seguir antes, durante y después de los juegos olímpicos.

De igual forma se puede compartir con discreción de buen uso a medios oficiales informativos para agendar reuniones, conferencias y demás eventos extras en donde los miembros del país puedan dar sus opiniones y experiencias.

7.14. Consulta 14

Atletas y Entrenadores menores o iguales a 30 años que practican o entrena Fútbol o Básquetbol.

```

    select nombre, primerApellido, segundoApellido, nombrePais, fechaNacimiento, nombreDisciplina
    from atleta natural join practicar
    where fechanacimiento > '1990-01-01' and (nombreDisciplina = 'Basketball' or nombreDisciplina = 'Football')
    union
    select nombre, primerApellido, segundoApellido, nacionalidad, fechaNacimiento, nombreDisciplina
    from entrenador
    where fechanacimiento > '1990-01-01' and (nombreDisciplina = 'Basketball' or nombreDisciplina = 'Football')
    order by nombreDisciplina, fechaNacimiento;
  
```

The screenshot shows a SQL query being run in a database environment. The query selects names, first and last names, country, birth date, and discipline for athletes and coaches born after 1990 who practice or train basketball or football. The results are displayed in a grid format, showing 31 records. The columns are labeled: Grid, Record, nombre, primerApellido, segundoApellido, nombrePais, fechanacimiento, and nombredisciplina. The data includes names like Selle, Luipold, Dyshart, Spain, 1990-10-08, Basketball, and various other names and details for both basketball and football disciplines across different countries and birth years.

Grid	Record	nombre	primerApellido	segundoApellido	nombrePais	fechanacimiento	nombredisciplina
1	1	Selle	Luipold	Dyshart	Spain	1990-10-08	Basketball
2	2	Torin	Durden	Laise	United States	1990-12-04	Basketball
3	3	Karla	Brunone	Axe	United States	1993-05-29	Basketball
4	4	Torrue	Kunisch	Bunford	United States	1993-06-17	Basketball
5	5	Solomon	Menelaws	Ceschelli	United States	1993-10-18	Basketball
6	6	Maje	Abdee	Fost	United States	1996-05-15	Basketball
7	7	Reginauld	Micklewright	Beavon	United States	1998-09-11	Basketball
8	8	Bell	Caldwall	Abrahamoff	United States	1998-10-16	Basketball
9	9	Farrell	McNelly	Treasaden	United States	1999-11-06	Basketball
10	10	Waldon	Whitemarsh	Trotbeck	Mexico	2000-03-28	Basketball
11	11	Rand	Dechelette	Spada	Japan	2002-05-11	Basketball
12	12	Haily	Cuthill	Moxley	United States	2002-11-10	Basketball
13	13	Chelsae	Grebert	McNelis	Spain	2003-01-11	Basketball
14	14	Laina	Cotes	Lumpkin	United States	2003-02-17	Basketball
15	15	Wini	Derycot	Geraldi	Japan	2003-06-08	Basketball
16	16	Robin	Kenwyn	Clapp	Japan	2003-06-11	Basketball
17	17	Web	Kalkhoven	Arias	Japan	2004-01-04	Basketball
18	18	Juanita	Ogilvy	Drillingcourt	Mexico	2004-02-07	Basketball
19	19	Aloysius	Shrubshall	Lyon	United States	2004-05-15	Basketball
20	20	Lianna	Crozier	Blackridge	Japan	2006-02-03	Basketball
21	21	Ty	Creser	Gumby	Mexico	2006-08-19	Basketball
22	22	Kellina	Masse	Benezet	United States	2007-03-21	Basketball
23	23	Obediah	Cowpland	Merton	United States	2007-07-21	Basketball
24	24	Lamond	Varley	Raffan	United States	2008-01-20	Basketball
25	25	Jecho	Lorens	Lillo	United States	2008-09-30	Basketball
26	26	Lane	Belliveau	Oland	Mexico	2009-01-03	Basketball
27	27	Priscella	Skitterel	Jovey	Mexico	2009-08-10	Basketball
28	28	Lorraine	Balcock	Cockhill	Japan	1992-03-11	Football
29	29	Guthrey	Lumby	Sparsholt	Spain	1992-04-25	Football
30	30	Cari	St Leger	Sheirlaw	Mexico	1992-07-31	Football
31	31	Arielle	Purselowe	Bottle	Mexico	1992-12-10	Football

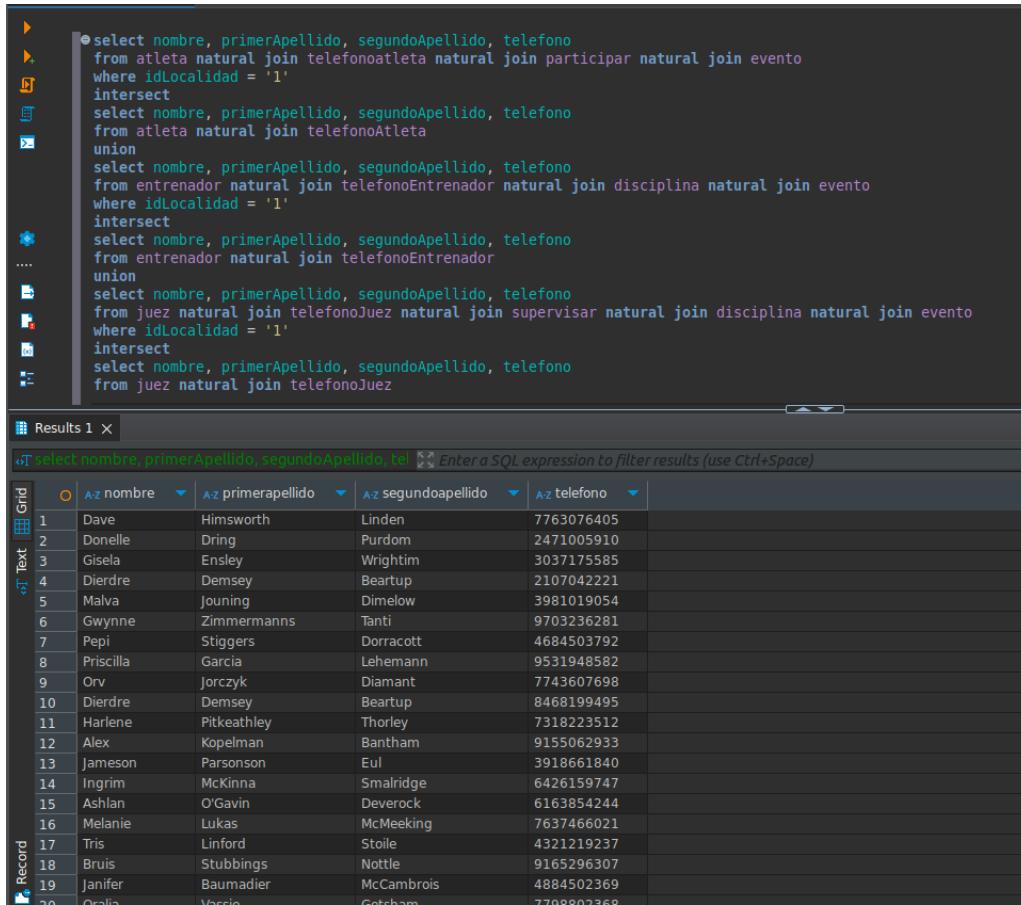
Figura 27: Consulta 14

El mundo del deporte a pesar de ser amplio, siempre ha tenido a sus favoritos, o más bien los más redituables. A veces la gente considera los eventos que involucran el fútbol y el básquetbol en los juegos olímpicos como una versión diluida de lo que puede llegar a ser la copa mundial o la NBA respectivamente, pero no se puede ignorar el hecho de que a pesar de eso sigue teniendo un gran impacto y genera curiosidad en los espectadores ya que pueden ver de primera mano una alineación diferente a lo que están acostumbrados.

Es por eso que el propósito de esta consulta, es buscar a aquellos atletas o entrenadores jóvenes de ambas disciplinas, ya que si su rendimiento cumple las expectativas podrían ser llamados a próximas sedes, o del lado de marketing, ser la imagen de vallas y spots publicitarios que llamen la atención de más personas para venir a Estados Unidos o sintonizar los juegos.

7.15. Consulta 15

El nombre completo y el teléfono de los Atletas, Entrenadores y Jueces que participaran, entrenaran o supervisaran en la localidad con id 1 "Los Angeles Memorial Coliseum"



```

    select nombre, primerApellido, segundoApellido, telefono
    from atleta natural join telefonoAtleta natural join participar natural join evento
    where idLocalidad = '1'
    intersect
    select nombre, primerApellido, segundoApellido, telefono
    from atleta natural join telefonoAtleta
    union
    select nombre, primerApellido, segundoApellido, telefono
    from entrenador natural join telefonoEntrenador natural join disciplina natural join evento
    where idLocalidad = '1'
    intersect
    select nombre, primerApellido, segundoApellido, telefono
    from entrenador natural join telefonoEntrenador
    union
    select nombre, primerApellido, segundoApellido, telefono
    from juez natural join telefonoJuez natural join supervisar natural join disciplina natural join evento
    where idLocalidad = '1'
    intersect
    select nombre, primerApellido, segundoApellido, telefono
    from juez natural join telefonoJuez
  
```

The screenshot shows a SQL query editor with a code pane containing a complex multi-table join query and a results pane below it. The results pane displays a grid of 20 records with columns: Grid, Record, nombre, primerApellido, segundoApellido, and telefono. The data includes names like Dave, Himsworth, Linden, and phone numbers like 7763076405.

Grid	Record	nombre	primerApellido	segundoApellido	telefono
1	1	Dave	Himsworth	Linden	7763076405
2	2	Donelle	Dring	Purdom	2471005910
3	3	Gisela	Ensley	Wrightim	3037175585
4	4	Dierdre	Demsey	Beartup	2107042221
5	5	Malva	Jouning	Dimelow	3981019054
6	6	Gwynne	Zimmermanns	Tanti	9703236281
7	7	Pepi	Stiggers	Dorracott	4684503792
8	8	Priscilla	Garcia	Lehemann	9531948582
9	9	Orv	Jorczyk	Diamant	7743607698
10	10	Dierdre	Demsey	Beartup	8468199495
11	11	Harlene	Pitkeathley	Thorley	7318223512
12	12	Alex	Kopelman	Bantham	9155062933
13	13	Jameson	Parsonson	Eul	3918661840
14	14	Ingrim	McKinna	Smalridge	6426159747
15	15	Ashlan	O'Gavin	Deverock	6163854244
16	16	Melanie	Lukas	McMeeking	7637466021
17	17	Tris	Linford	Stolle	4321219237
18	18	Bruis	Stubblings	Nottle	9165296307
19	19	Janifer	Baumadier	McCambrois	4884502369
20	20	Oralia	Vessina	Cathelem	7700002360

Figura 28: Consulta 14

El propósito de esta consulta, la cual puede ser aplicada a cualquier otra localidad es poder tener al alcance la información más básica de las personas participantes de esa cierta localidad. Como administradores del evento es sumamente importante esto por cualquier emergencia o aviso urgente relacionado al lugar en el que estarán, ya que se podría mandar un mensaje de texto o llamada automática de manera instantánea.

También puede servir como una lista reducida y de fácil contacto con los participantes de tal localidad por otros motivos más específicos.