



# Tecnológico de Monterrey

## Ejercicio de Modelación de Base de Datos y creación de scripts de SQL

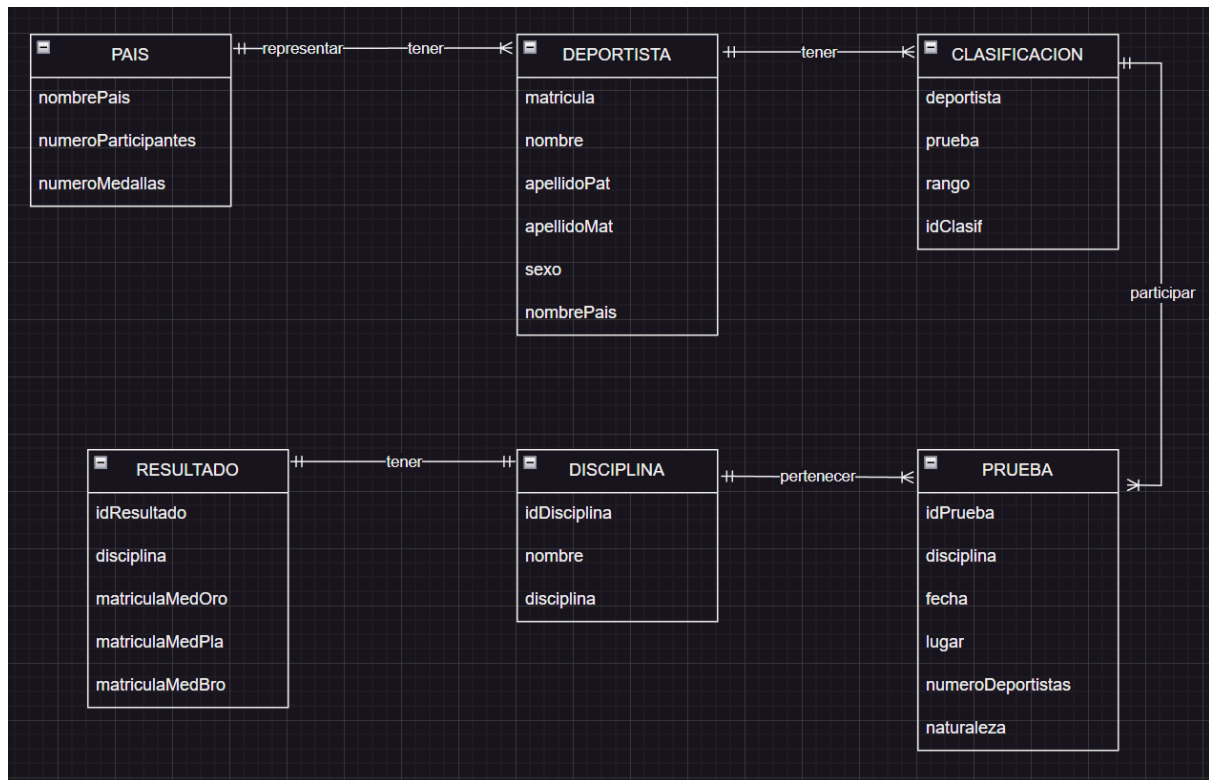
Angel Adrian Afonso Castellanos - A01782545

20 de mayo del 2023

Construcción de Software y Toma de decisiones.

Dr. Esteban Castillo Juárez

## Esquema Relacional



## Relaciones

- Un país puede tener mínimo uno o varios deportistas. Un deportista puede representar máximo solo a un país. (1:1..N).
- Un deportista puede tener mínimo una o varias clasificaciones. Una clasificación sólo puede tener un deportista. (1:1..N).
- Una clasificación puede tener mínimo una o varias pruebas. Una prueba puede tener máximo solo una clasificación. (1:1..N).
- Una prueba puede tener solo una disciplina. Una disciplina puede tener mínimo una o varias pruebas. (1:1..N).
- Una disciplina puede tener solo un resultado. Un resultado puede tener máximo solo una disciplina. (1:1).

## Atributos

| PAIS |                     |                    |   |
|------|---------------------|--------------------|---|
| KEY  | ATRIBUTO            | DESCRIPCIÓN FÍSICA | DESCRIPCIÓN SEMÁNTICA                   |
| PK   | nombrePais          | str                | contiene el nombre del país.            |
|      | nombreParticipantes | int                | número de participantes de cada país.   |
|      | numeroMedallas      | int                | número de medallas ganadas por el país. |

| DEPORTISTA |             |                    |  |
|------------|-------------|--------------------|--|
| KEY        | ATRIBUTO    | DESCRIPCIÓN FÍSICA | DESCRIPCIÓN SEMÁNTICA  |
| PK         | matricula   | str                | conjunto ordenado de caracteres que identifican a los deportistas. |
|            | nombre      | str                | almacena el nombre del deportista.                                 |
|            | apellidoPat | str                | almacena el apellido paterno del deportista.                       |
|            | apellidoMat | str                | almacena el apellido materno del deportista.                       |
|            | sexo        | char(1)            | almacena la letra que representa el sexo del deportista (M,F).     |
| FK         | nombrePais  | str                | nombre del pais al que pertenece el deportista.                    |

| CLASIFICACION |            |                    |                                       |
|---------------|------------|--------------------|---------------------------------------|
| KEY           | ATRIBUTO   | DESCRIPCIÓN FÍSICA | DESCRIPCIÓN SEMÁNTICA                 |
| FK            | deportista | str                | almacena la matrícula del deportista. |

|    |          |         |  |
|----|----------|---------|--|
| FK | prueba   | str     | identificador de la prueba.                    |
|    | rango    | char(1) | almacena la posición del deportista            |
| PK | idClasif | str     | almacena el identificador de la clasificación. |

| PRUEBA |                   |                    |  |
|--------|-------------------|--------------------|--|
| KEY    | ATRIBUTO          | DESCRIPCIÓN FÍSICA | DESCRIPCIÓN SEMÁNTICA                              |
| PK     | idPrueba          | str                | identificador de prueba                            |
| FK     | disciplina        | str                | disciplina a la que pertenece la prueba            |
|        | fecha             | date               | fecha de la prueba                                 |
|        | lugar             | str                | lugar de la prueba                                 |
|        | numeroDeportistas | int                | cantidad de deportistas participantes en la prueba |
|        | naturaleza        | str                | almacena el tipo de prueba(eliminatoria, final)    |

| DISCIPLINA |              |                    |                             |
|------------|--------------|--------------------|-----------------------------|
| KEY        | ATRIBUTO     | DESCRIPCIÓN FÍSICA | DESCRIPCIÓN SEMÁNTICA       |
| PK         | idDisciplina | str                | identificador de disciplina |
|            | nombre       | str                | nombre de la disciplina     |
| FK         | disciplina   | str                | categoría de la disciplina  |

| RESULTADO |          |                    |                       |
|-----------|----------|--------------------|-----------------------|
| KEY       | ATRIBUTO | DESCRIPCIÓN FÍSICA | DESCRIPCIÓN SEMÁNTICA |

|    |                 |     |  |
|----|-----------------|-----|--|
| PK | idResultado     | str | identificador de resultado                               |
| FK | disciplina      | str | almacena la disciplina perteneciente                     |
|    | matriculaMedOro | str | matrícula del deportista acreedor a la medalla de oro    |
|    | matriculaMedPla | str | matrícula del deportista acreedor a la medalla de plata  |
|    | matriculaMedBro | str | matrícula del deportista acreedor a la medalla de bronce |

El esquema relacional presentado cumple con la tercera forma normal ya que un requisito de ella es cumplir de igual forma con la primera y segunda forma normal, las cuales se cumplen y las tablas contienen una única llave primaria y algunas contienen llaves foráneas.

Una de las características que se necesita cumplir en el esquema para que este sea normalizado en la primera forma es que todos los atributos deben ser atómicos, lo que significa que son indivisibles. Se tuvo que dividir el atributo apellidos en DEPORTISTA por dos atributos, uno para cada apellido. De igual manera para cumplir con la primera forma normal, todos los atributos son monovalentes, es decir que almacenan un único valor.

Ya teniendo cumplidos los requisitos de la 1FN, se puede hablar un poco de la 2FN, la cual pide que los atributos no sean dependientes completamente de la llave, cosa que se cumple en el esquema relacional. Y finalmente, la 3FN también se cumple debido a que los atributos no clave no son dependientes de manera transitiva de la llave primaria. Por lo que se puede afirmar que el esquema relacional presentado cumple con la 3FN.

## **Códigos de mySQL**

Apellidos y nombre de los participantes de nacionalidad mexicana:

```
SELECT apellidoPat, apellidoMat, nombre  
FROM DEPORTISTA  
WHERE nombrePais = 'mexico';
```

Apellidos, nombre y puntos acumulados de los participantes de USA:

```
SELECT apellidoPat, apellidoMat, nombre, rango  
FROM DEPORTISTA  
WHERE nombrePais = 'USA';
```

Apellidos y nombre de los participantes que se clasificaron en primer lugar en al menos una competencia:

```
SELECT apellidoPat, apellidoMat, nombre  
FROM DEPORTISTA  
WHERE idClasif IN (  
    SELECT idClasif  
    FROM CLASIFICACION  
    WHERE rango = 1  
);
```

Nombre de las competencias en las que intervinieron los participantes mexicanos:

```
SELECT prueba  
FROM CLASIFICACION  
WHERE idClasif IN (  
    SELECT idPrueba  
    FROM PRUEBA  
    WHERE deportista IN (  
        SELECT deportista  
        FROM CLASIFICACION  
        WHERE nombrePais = 'mexico'
```

```
)  
);
```

Apellidos y nombre de los participantes que nunca se clasificaron en primer lugar en alguna competencia:

```
SELECT apellidoPat, apellidoMat, nombre  
FROM DEPORTISTA  
WHERE matricula NOT IN (  
    SELECT deportista  
    FROM CLASIFICACION  
    WHERE rango = 1  
);
```

Apellidos y nombre de los participantes que siempre se clasificaron en alguna competencia:

```
SELECT apellidoPat, apellidoMat, nombre  
FROM DEPORTISTA  
WHERE matricula NOT IN (  
    SELECT matricula  
    FROM CLASIFICACION  
    WHERE rango <> 1  
);
```

Nombre de la competencia que aporta el máximo de puntos:

```
SELECT nombre  
FROM DISCIPLINA  
WHERE idDisciplina = (  
    SELECT idPrueba  
    FROM PRUEBA  
    GROUP BY disciplina
```

```
ORDER BY SUM(idResultado) DESC
LIMIT 1
);
```

Países (nacionalidades) que participaron en todas las competencias:

```
SELECT nombrePais
FROM DEPORTISTA
GROUP BY nombrePais
HAVING COUNT(DISTINCT idPrueba) = (
    SELECT COUNT(DISTINCT idPrueba)
    FROM PRUEBA
);
```

Propuesta de consulta con una sola tabla y una función (MIN, AVG, etc.):

```
SELECT MIN(rango) AS PuntosMinimos
FROM CLASIFICACION;
```

Propuesta de consulta que involucre dos tablas con GROUP BY:

```
SELECT DEPORTISTA.Nombre, AVG(CLASIFICACION.rango) AS PuntosPromedio
FROM DEPORTISTA P
JOIN CLASIFICACION C ON P.matricula = C.matricula
GROUP BY P.matricula;
```

Propuesta de consulta que involucre tres tablas con LEFT JOIN, ORDER BY, GROUP BY y LIMIT:

Aqui si fallé

Propuesta de consulta que involucre tres tablas con LEFT JOIN, ORDER BY, GROUP BY y LIMIT:

aqui tambien porque tuve problemas con mi compu y lo tuve que hacer desde otra que no tiene mySQL asi que las de arriba estan hechas con como creo que quedaria en mySQL.