

Primera forma normal

La idea de esta forma es que no haya datos repetitivos y cada clave sea única.

Ninguna tabla de las presentes tiene datos repetitivos, esto para hacer las consultas más sencillas y evitar tablas redundantes. También es importante resaltar que cada una de ellas cuenta con una sola clave única, esto es más sencillo para el administrador que tiene que comunicar las consultas correctamente.

Segunda forma normal

Para que se cumpla no hay atributos no clave que no dependan de alguno clave.

En cada atributo que no se encuentra en la llave primaria da un hecho que depende de la llave completa. No se encuentran atributos de llave no primaria que sean hechos sobre un subconjunto.

Tercera forma normal

Por último las llaves foráneas permiten la relación entre entidades.

Todas las entidades dentro del esquema relacional cumplen la tercera forma normal. Ningún dato no clave, es decir, que no son llaves primarias, están presentes en otras entidades a las cuales no corresponde. Están garantizando la integridad en el esquema relacional.

Por otra parte, las llaves foráneas que pueden estar en otras entidades, esto debido a que hace referencia a diferentes temas, así pudiendo relacionar entidades.

La mayoría de estas formas es para evitar redundancia, así que se quitó la tabla de resultados. El punto de esta tabla es obtener las medallas de cada deportista. Sin embargo, analizando la tabla XXXXXXXX podemos ver que esos mismos datos se encuentran ahí.

El punto de una herramienta como MySQL es facilitar las consultas, así que usándola correctamente podemos presentar las mismas consultas utilizando otras tablas. Por lo tanto, omitir la tabla de resultados evita redundancias.

1. Apellidos y nombre de los participantes de nacionalidad mexicana.

```
SELECT Apellido, nombre  
FROM Deportistas  
WHERE IdPais = 'mexicana';
```

2. Apellidos, nombre y puntos acumulados de los participantes de USA.

```
SELECT Apellido, nombre, puntaje  
FROM puntos  
INNER JOIN participante ON puntos.id_participante = participante.id  
WHERE participante.IdPais= 'USA';
```

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

```
SELECT Apellido, nombre
FROM participantes
INNER JOIN primeros_lugares ON participantes.id = primeros_lugares.id_participante;
```

4. **Nombre de las competencias en las que intervinieron los participantes mexicanos.**

```
SELECT *
FROM clasificacion
INNER JOIN participante ON clasificacion.id_participante = participante.id
WHERE participante.nacionalidad = 'mexicana';
```

5. **Apellidos y nombre de los participantes que nunca se clasificaron en primer lugar en alguna competencia.**

```
SELECT apellido, nombre
FROM participantes
WHERE id NOT IN
(
SELECT participantes.id
FROM participantes
INNER JOIN clasificacion ON participantes.id = clasificacion.id_participante
WHERE lugar = 1
)
```

6. **Apellidos y nombre de los participantes siempre se clasificaron en alguna competencia.**

En este caso la consulta se puede interpretar de una variedad de maneras, en este caso se va a interpretar como las personas que participaron en alguna competencia. Por ende, solo es necesario obtener los nombres de las personas que están enlistadas en algún deporte, el que sea.

```
SELECT apellido, nombre
FROM clasificacion
INNER JOIN participantes ON clasificacion.id_participante = participantes.id;
```

7. **Nombre de la competencia que aporta el máximo de puntos.**

```
SELECT Nombre
FROM competencia
INNER JOIN
(
```

```
SELECT num_pto  
FROM competencia  
WHERE num_pto1 < num_pto2  
)
```

8. Países (nacionalidades) que participaron en todas las competencias.

En este caso también es muy ambiguo, la gramática se puede de interpretar de dos maneras

Opción 1: Hace referencia a los países que participaron, países que calificaron o participaron, por ende, los países que tienen alguna persona en algún deporte.

Opción 2: Si es que hay algún país que participo en todas las competencias (en el ejemplo es el país 3) estamos buscando este país.

La siguiente explicación es para la opción 1.

```
SELECT nombre_pais  
FROM Id_pais
```

9. Porpongan una consulta que involucre dos tabla con GROUP BY --- Nueva consulta

```
SELECT p.nombre AS Pais, p.NumParticipantes, AVG(c.PuntuajeTotal) AS  
PromedioPuntuaje  
FROM olimpiadas.Pais p  
JOIN olimpiadas.Deportistas d ON p.idpais = d.IdPais  
JOIN olimpiadas.Clasificacion c ON d.Matricula = c.Matricula  
GROUP BY p.idpais, p.nombre, p.NumParticipantes;
```

Conclusión

Espero poder seguir trabajando esto más para entender bien las bases de datos.