



Tecnológico de Monterrey

Ejercicio de Modelación de Base de Datos y Creación de Scripts

Alumno:

Juan Pablo Cruz Rodríguez A01783208

Docente:

Esteban Castillo Juarez

Campus Santa Fe

20 Mayo 2023

El primer paso es la interpretación de la información administrada para la creación de tablas relacionales, las cuales deben de pasar por un proceso de análisis y normalización para facilitar las búsqueda de datos dentro de la base.

A continuación se observa la información que fue compartida al igual que el contexto para la creación:

1. PAÍS: nombre, número de participantes, número de medallas.
2. DEPORTISTA: matrícula, nombre, apellidos, sexo, país.
3. DISCIPLINA: identificador, nombre (p. ej. 400M nado libre), disciplina (p. ej. natación).
4. PRUEBA: identificador, disciplina, fecha, lugar, número de deportistas inscritos, naturaleza (eliminatória, final). Para cada disciplina hay varias pruebas eliminatorias para una sola final.
5. CLASIFICACIÓN: deportista, prueba, rango (p. ej. 1, 2, ...).
6. RESULTADO: disciplina, matrícula del deportista con medalla de oro, matrícula del deportista con medalla de plata, matrícula del deportista con medalla de bronce.

De acuerdo a la información de la tablas y sus atributos se visualiza las siguientes tablas:

Pais	
PK	<u>Id Pais INT NOT NULL UQ AI</u>
	nombre VARCHAR(45) NOT NULL UQ
	num_participantes INT NOT NULL
	num_medallas INT NOT NULL

Deportista	
PK	<u>Matricula INT NOT NULL UQ</u>
	Nombre VARCHAR(45) NOT NULL
	Apellido VARCHAR(45) NOT NULL
	Sexo VARCHAR(45) NOT NULL
	Pais VARCHAR(45) NOT NULL

Disciplina	
PK	<u>Id Disciplina INT NOT NULL UQ AI</u>
	nombre_disciplina VARCHAR(45) NOT NULL
	disciplina VARCHAR(45) NOT NULL

Prueba	
PK	<u>Id Prueba INT NOT NULL UQ AI</u>
FK	Disciplina VARCHAR(45) NOT NULL
	Fecha DATE NOT NULL
	Lugar VARCHAR(45) NOT NULL
	Numero_Participantes INT NOT NULL
	Naturaleza VARCHAR(45) NOT NULL

Clasificacion	
PK	<u>Deportista VARCHAR(45) NOT NULL UQ</u>
	Prueba VARCHAR(45) NOT NULL
	Rango INT NOT NULL

Resultado	
	<u>Disciplina VARCHAR(45) NOT NULL</u>
	Matricula_oro INT NOT NULL UQ
	Matricula_plata INT NOT NULL UQ
	Matricula_bronces INT NOT NULL UQ

A partir de la creación de las tablas anteriores, debemos de normalizarlas para evitar datos repetidos, ambigüedades y caos en la información. De hecho, la normalización nos permite desglosar la información para mejorar la compresión de cada uno de los atributos y registros de las tablas.

Las formas de normalización que utilizamos son: 1f, 2f y 3f normal:

1. Primera Forma Normal (1NF):

- Los valores en una tabla deben ser atómicos, es decir, no deben contener datos repetidos o grupos de datos.
- Cada columna debe contener un solo valor.
- No debe haber una dependencia funcional transitiva, lo que significa que los atributos no clave no deben depender de otros atributos no clave.

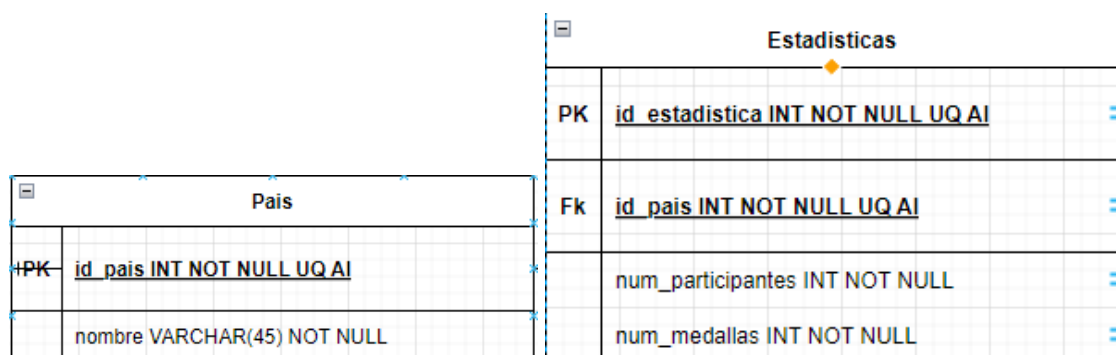
2. Segunda Forma Normal (2NF):

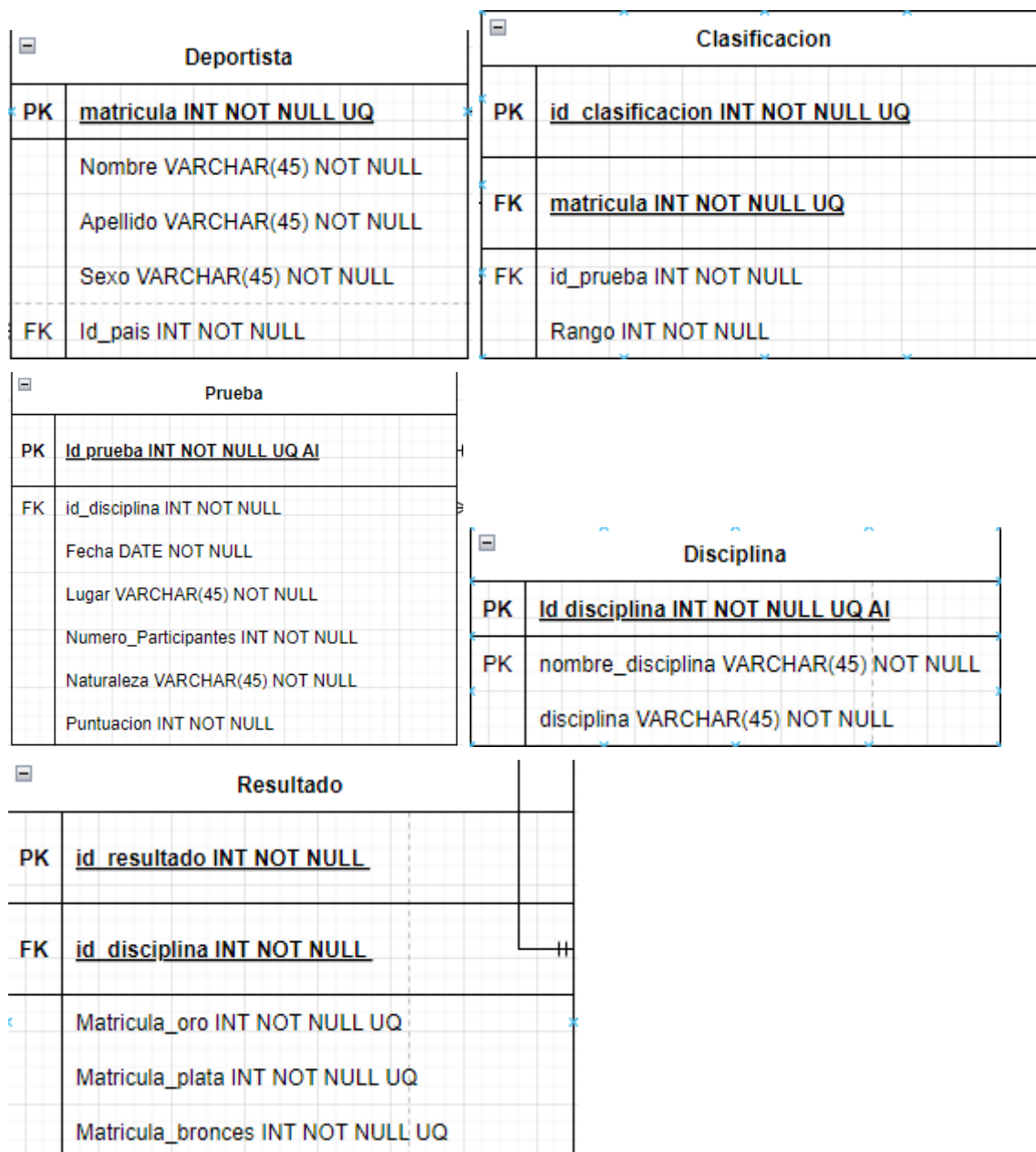
- La tabla debe cumplir con 1NF.
- Todos los atributos no clave deben depender completamente de la clave primaria.
- Si la tabla tiene una clave primaria compuesta, cada atributo no clave debe depender de la clave completa y no solo de una parte de ella.

3. Tercera Forma Normal (3NF):

- La tabla debe cumplir con 2NF.
- No debe haber dependencias funcionales transitivas, lo que significa que los atributos no clave no deben depender de otros atributos no clave.
- Los atributos no clave deben depender únicamente de la clave primaria.

Siguiendo los pasos de cada forma normal y con base a la interpretación del creador de la base de datos, se obtuvo las siguientes tablas normalizadas justificadas en la tercera forma, al igual que el diagrama de entidad relacion con sus respectivos cardinalidades:





En el diagrama de entidad relación identifique en total 7 entidades y 6 relaciones las cuales son las siguientes:

- Entidades:
 - País: Esta tabla muestra el nombre del país.
 - Estadística: Esta tabla posee los datos de cada país.
 - Deportista: Esta tabla muestra los datos de cada deportista.
 - Clasificación: Esta tabla posee información acerca del rango en el que quedó el deportista.
 - Prueba: Esta tabla muestra la información de la prueba que se llevará a cabo.
 - Disciplina: Esta tabla posee las diferentes categorías o deportes.
 - Resultado: Esta tabla muestra los resultados de la disciplina.

- Relaciones:
 - (País) 1:1 (Estadística):
 - Restricciones de integridad:
 - Pk id pais : Llave primaria que indica el país
 - Pk id estadística: Llave primaria que indica la estadística
 - Fk id pais: Llave foránea en estadística que apunta a la tabla país
 - (País) 1:N (Deportista):
 - Restricciones de integridad:
 - Pk id pais : Llave primaria que indica el país
 - Pk matrícula: Llave primaria que identifica al deportista
 - Fk id pais: Llave foránea en estadística que apunta a la tabla país
 - (Deportista) 1:1 (Clasificación):
 - Restricciones de integridad:
 - Pk matrícula: Llave primaria que identifica al deportista
 - Pk id clasificación: Llave primaria que indica a la clasificación de cada deportista
 - Fk matrícula: Llave foránea en clasificación que apunta a la tabla deportista
 - Fk id prueba: Llave foránea en clasificación que apunta a la tabla prueba
 - Fk id pais: Llave foránea en deportista que apunta a la tabla país
 - (Clasificación) N:1 (Prueba):
 - Restricciones de integridad:
 - Pk id clasificación: Llave primaria que indica a la clasificación de cada deportista
 - Pk id prueba: Llave foránea que identifica cada registro de la prueba
 - Fk matrícula: Llave foránea en clasificación que apunta a la tabla deportista
 - Fk id prueba: Llave foránea en clasificación que apunta a la tabla prueba
 - Fk id disciplina: Llave foránea en prueba que apunta a la tabla disciplina
 - (Prueba) N:1 (Disciplina):
 - Restricciones de integridad:
 - Pk id prueba: Llave foránea que identifica cada registro de la prueba
 - Pk id disciplina: Llave primaria que muestra cada disciplina
 - Pk nombre disciplina: Llave primaria que muestra la sub-disciplina o la categoría
 - Fk id disciplina: Llave foránea en prueba que apunta a la tabla disciplina
 - (Disciplina) N:1 (Resultado):
 - Restricciones de integridad:

- Pk id disciplina: Llave primaria que muestra cada disciplina
- Pk nombre disciplina: LLave primaria que muestra la sub-disciplina o la categoría
- Pk id resultado: LLave primaria que identifica cada resultado
- Fk id disciplina: Llave foránea en resultado que apunta a la tabla disciplina

En cada una de las tablas normalizadas utilizamos la restricción de datos o restricción de integridad lo que permite una mejor coherencia y consistencia entre las tablas, al igual que en sus relaciones. Estas restricciones se demuestran a través de las llaves primarias y foráneas, las cuales permiten eliminar datos redundantes y conectan las diferentes tablas dentro del SGBD.

Las restricciones de integridad y la normalización de tercera forma se presentan en el diagrama de entidad relación, ya que no se presentan ningún registro en algunas de las tablas con valores o datos ambiguos o repetidos causando confusión en el análisis de la información. Cada tabla posee datos atómicos y específicos por lo que se elimina posibles errores dentro de las tablas.

Una vez creado y analizado nuestro esquema o diagrama, nos dirigimos a MySQL para crear 10 registros por tabla y realizar las siguientes consultas para verificar la unicidad e integridad de nuestros datos:

Script:

```
-- Comando para entrar al esquema
USE tarea_modelacion_base_de_datos;
-- Tabla con codigo 1
CREATE TABLE pais(
id_pais int primary key not null unique auto_increment comment "Id del pais",
nombre varchar(45) not null comment "Nombre del pais");
-- Tabla con codigo 2
CREATE TABLE estadistica(
id_estadistica int primary key not null unique auto_increment comment "Id de la estadistica",
id_pais int not null comment "Llave foranea del id del pais",
num_participantes int not null comment "Total de participantes por pais",
num_medallas int not null comment "Total de medallas por pais",
CONSTRAINT fk_pais FOREIGN KEY (id_pais) REFERENCES pais (id_pais));
-- Tabla con codigo 3
CREATE TABLE deportista(
matricula int primary key not null unique comment "Matricula del deportista",
id_pais int not null comment "LLave foranea del id del pais",
nombre varchar(45) not null comment "Nombre del deportista",
apellido varchar(45) not null comment "Apellido del deportista",
sexo varchar(45) not null comment "Sexo del deportista",
CONSTRAINT fk_pais_deportista FOREIGN KEY (id_pais) REFERENCES pais (id_pais));
-- Tabla con codigo 4
CREATE TABLE clasificacion(
```

```

id_clasificacion int primary key not null unique comment "Id de la clasificacion",
matricula int not null comment "LLave foranea matricula del deportista",
id_prueba int not null comment "LLave foranea de la prueba",
rango int not null comment "Lugar del participante",
CONSTRAINT fk_matricula FOREIGN KEY (matricula) REFERENCES deportista
(matricula),
CONSTRAINT fk_prueba FOREIGN KEY (id_prueba) REFERENCES prueba (id_prueba));
-- Tabla con codigo 5
CREATE TABLE prueba(
id_prueba int primary key not null unique auto_increment comment "Id de la prueba",
id_disciplina int not null comment "LLave foranea de la disciplina",
fecha date not null comment "Fecha de la prueba",
lugar varchar(45) not null comment "Lugar de la competencia",
numero_participantes int not null comment "Numero de deportistas inscritos en esta
prueba",
naturaleza varchar(45) not null comment "Naturaleza de la prueba eliminatoria, final directa
o ambas",
puntuacion int not null comment "Puntuacion que da la prueba",
CONSTRAINT fk_disciplina FOREIGN KEY (id_disciplina) REFERENCES disciplina
(id_disciplina));
-- Tabla con codigo 6
CREATE TABLE disciplina(
id_disciplina int not null unique auto_increment comment "LLave primaria compuesta Id de
la disciplina",
nombre_disciplina varchar(45) not null comment "LLave primaria compuesta Nombre de la
disciplina en especifico",
disciplina varchar(45) not null comment "Disciplina general",
PRIMARY KEY (id_disciplina,nombre_disciplina));
-- Tabla con codigo 7
CREATE TABLE resultado(
id_resultado int primary key not null unique auto_increment comment "Id del resultado",
id_disciplina int not null comment "LLave foranea de la disciplina",
matricula_oro int not null unique comment "Matricula del deportista que gano medalla de
oro",
matricula_plata int not null unique comment "Matricula del deportista que gano medalla de
plata",
matricula_bronce int not null unique comment "Matricula del deportista que gano medalla de
bronce",
CONSTRAINT fk_disciplina_resultado FOREIGN KEY (id_disciplina) REFERENCES
disciplina (id_disciplina));

-- Ingresamos informacion en cada una de las tablas (10 registros)

-- Tabla pais
SET AUTOCOMMIT=0;
INSERT INTO pais VALUES (1,"Mexico"),
(2,"Colombia"),
(3,"USA"),

```

```
(4,"España"),
(5,"Argentina"),
(6,"Alemania"),
(7,"Rusia"),
(8,"Japon"),
(9,"Francia"),
(10,"Egipto");
COMMIT;
```

-- Tabla Estadistica

```
SET AUTOCOMMIT=0;
INSERT INTO estadistica VALUES (1,8,40,57),
(2,10,30,6),
(3,2,10,5),
(4,3,120,112),
(5,5,20,3),
(6,7,80,71),
(7,6,77,37),
(8,9,68,33),
(9,4,50,17),
(10,1,28,4);
COMMIT;
```

-- Tabla Deportista

```
SET AUTOCOMMIT=0;
INSERT INTO deportista VALUES (8320,1,"Camila","Lopez","Femenino"),
(2783,3,"Miles","Morales","Masculino"),
(6382,1,"Diego","Perez","Masculino"),
(7391,1,"Alexander","Cruz","Masculino"),
(1031,1,"Daniel","Garcia","Masculino"),
(7301,3,"Robert","Patterson","Masculino"),
(8302,8,"Tanaka","Shoyo","Femenino"),
(5201,2,"Juana","Valentina","Femenino"),
(1028,9,"David","Leroy","Masculino"),
(0183,4,"Sofia","Rodriguez","Femenino");
COMMIT;
```

-- Tabla Clasificacion

```
SET AUTOCOMMIT=0;
INSERT INTO clasificacion VALUES (1,8320,2,10),
(2,2783,6,7),
(3,6382,9,3),
(4,7391,8,6),
(5,1031,1,15),
(6,7301,3,8),
(7,8302,7,4),
(8,5201,5,1),
(9,1028,4,1),
```



```
(10,0183,10,2);  
COMMIT;
```

-- Tabla Prueba

```
SET AUTOCOMMIT=0;  
INSERT INTO prueba VALUES (1,10,"2021-08-2","Tokio",30,"eliminatória",100),  
(2,9,"2021-08-6","Kyoto",40,"eliminatória",50),  
(3,8,"2021-08-10","Tokio",20,"final",70),  
(4,7,"2021-08-19","Osaka",40,"eliminatória",80),  
(5,6,"2021-08-23","Kyoto",60,"final",30),  
(6,5,"2021-08-4","Nagoya",50,"eliminatória",20),  
(7,4,"2021-08-18","Tokio",15,"final",40),  
(8,3,"2021-08-15","Nagoya",25,"eliminatória",90),  
(9,2,"2021-08-10","Nagoya",35,"eliminatória",60),  
(10,1,"2021-08-2","Nagoya",30,"eliminatória",10);  
COMMIT;
```

-- Tabla Disciplina

```
SET AUTOCOMMIT=0;  
INSERT INTO disciplina VALUES (1,"playa","voleibol"),  
(2,"rapido","futbol"),  
(3,"500 mts","atletismo"),  
(4,"400 mts","natacion"),  
(5,"sala","voleibol"),  
(6,"11vs11","futbol"),  
(7,"larga distancia","tiro con arco"),  
(8,"con vallas","atletismo"),  
(9,"de mesa","tenis"),  
(10,"sincronizado","natacion");  
COMMIT;
```

-- Tabla Resultado

```
SET AUTOCOMMIT=0;  
INSERT INTO resultado VALUES(1,2,6382,8201,5678),  
(2,4,9081,9758,6451),  
(3,6,5201,8250,4892),  
(4,8,8752,6725,9872),  
(5,10,0789,9875,9827),  
(6,1,0183,8200,0986),  
(7,3,9821,7625,0672),  
(8,5,9871,5467,8761),  
(9,7,1028,5632,0922),  
(10,9,7842,5380,6573);  
COMMIT;
```

-- Updates

```
UPDATE deportista  
SET puntos_acumulados = FLOOR(1 + RAND()*100)
```

LIMIT 10;

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

Script:

```
-- Apellidos y nombres de los participantes con nacionalidad mexicana
SELECT apellido,nombre FROM tarea_modelacion_base_de_datos.deportista WHERE
id_pais = 1;
```

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

Script:

```
-- Apellidos,nombres y puntos acumulados de los participantes con nacionalidad USA
SELECT                apellido,nombre,puntos_acumulados                FROM
tarea_modelacion_base_de_datos.deportista WHERE id_pais = 3;
```

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

Script:

```
-- Apellidos y nombre de los deportistas que se clasificaron en primer lugar en al menos una
competencia
SELECT apellido,nombre FROM tarea_modelacion_base_de_datos.deportista INNER JOIN
tarea_modelacion_base_de_datos.clasificacion
ON deportista.matricula = clasificacion.matricula
WHERE clasificacion.rango = 1;
```

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

Script:

```
-- Nombre de las competencias en las que intervinieron los participantes mexicanos
SELECT disciplina,nombre_disciplina FROM tarea_modelacion_base_de_datos.disciplina
INNER JOIN tarea_modelacion_base_de_datos.prueba ON disciplina.id_disciplina =
prueba.id_disciplina
INNER JOIN tarea_modelacion_base_de_datos.clasificacion ON clasificacion.id_prueba =
prueba.id_prueba
INNER JOIN tarea_modelacion_base_de_datos.deportista ON deportista.matricula =
clasificacion.matricula
WHERE deportista.id_pais = 1;
```

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

Script:

-- Apellido y nombre de los participantes que nunca se clasificaron en primer lugar en alguna competencia
 SELECT apellido,nombre FROM tarea_modelacion_base_de_datos.deportista INNER JOIN
 tarea_modelacion_base_de_datos.clasificacion
 ON deportista.matricula = clasificacion.matricula
 WHERE NOT clasificacion.rango = 1;

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

Script:

-- Apellido y nombre de los participantes siempre se clasificaron en alguna competencia
 (Tomo como clasificados el top 10)
 SELECT apellido,nombre FROM tarea_modelacion_base_de_datos.deportista INNER JOIN
 tarea_modelacion_base_de_datos.clasificacion
 ON deportista.matricula = clasificacion.matricula
 WHERE NOT clasificacion.rango > 11;

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

Script:

-- Nombre de la competencia que aporta el maximo de puntos
 SELECT disciplina,nombre_disciplina FROM tarea_modelacion_base_de_datos.disciplina
 INNER JOIN tarea_modelacion_base_de_datos.prueba ON prueba.id_disciplina =
 disciplina.id_disciplina
 WHERE prueba.puntuacion = (SELECT MAX(puntuacion) FROM prueba);

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

Script:

-- Países que participaron en todas las competencias
 SELECT nombre FROM tarea_modelacion_base_de_datos.pais
 WHERE NOT EXISTS (SELECT id_disciplina FROM disciplina
 WHERE NOT EXISTS (SELECT * FROM deportista WHERE
 deportista.id_pais=pais.id_pais AND deportista.matricula IN
 (SELECT matricula FROM clasificacion WHERE clasificacion.id_prueba IN (
 SELECT id_prueba FROM prueba WHERE prueba.id_disciplina =
 disciplina.id_disciplina))));

9. Propongan una consulta que involucre una sola tabla con alguna función como MIN, AVG --- Nueva consulta

Script:

-- Consulta Extra:

-- El mínimo número de puntos que otorga una competencia
SELECT MIN(puntuacion) FROM tarea_modelacion_base_de_datos.prueba;

10. Propongan una consulta que involucre dos tabla con GROUP BY --- Nueva consulta

Script:

-- Consulta Extra:
-- Deportistas agrupados por su rango
SELECT deportista.nombre, deportista.apellido, clasificacion.rango FROM deportista
INNER JOIN tarea_modelacion_base_de_datos.clasificacion ON clasificacion.matricula =
deportista.matricula
GROUP BY nombre,apellido,rango;

11. Propongan una consulta que involucre tres tablas con las sentencias LEFT JOIN,
ORDER BY, GROUP BY Y LIMIT--- Nueva consulta

Script:

-- Consulta Extra:
-- Los primeros cinco nombres y apellidos de deportistas que clasificaron arriba del top 5 y
juegan en tokio
SELECT nombre,apellido FROM deportista
LEFT JOIN tarea_modelacion_base_de_datos.clasificacion ON clasificacion.matricula =
deportista.matricula
LEFT JOIN tarea_modelacion_base_de_datos.prueba ON prueba.id_prueba =
clasificacion.id_prueba
WHERE rango > 5
GROUP BY nombre,apellido,lugar
ORDER BY nombre
LIMIT 5;

12. Propongan una consulta que involucre tres tablas con las sentencias INNER JOIN y
LIKE

Script:

-- Consulta Extra:
-- Nombre y apellido de los deportistas que participaron en disciplina de sala
SELECT nombre, apellido, disciplina.nombre_disciplina
FROM deportista
INNER JOIN clasificacion ON deportista.matricula = clasificacion.matricula
INNER JOIN prueba ON clasificacion.id_prueba = prueba.id_prueba
INNER JOIN disciplina ON prueba.id_disciplina = disciplina.id_disciplina
WHERE nombre_disciplina LIKE '%sala%';

Las cuatro consultas extras son importantes debido a que:

- Primera: Saber cual es el mínimo de puntos que puede ganar en una competencia.
- Segunda: Ver el nombre y el rango de los deportistas de manera rápida.
- Tercera: Conocer los jugadores que se clasificaron dentro del top 5 y jugarán en Tokio.
- Cuarta: Ver qué jugadores participaron en alguna disciplina en sala.