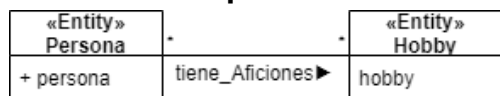
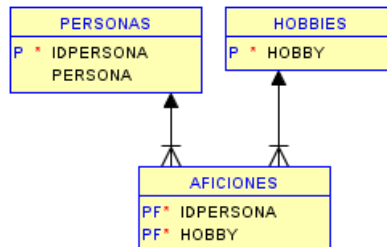


Modelo conceptual UML



IDPERSONA	PERSONA
1	Ana
2	Pedro
3	Joaquin
4	Luisa
5	Amorfo

Base de datos Relacional



HOBBY
Cine
Lectura
Fútbol
Tenis
Senderismo

IDPERSONA	HOBBY
1	Cine
2	Lectura
2	Fútbol
3	Tenis
3	Senderismo
4	Cine
4	Lectura
4	Fútbol
4	Tenis
4	Senderismo

personas(idPersona:integer,persona:char)

PK(idPersona)

Hobbies(hobby:char)

PK(hobby)

Aficiones(idPersona:persona,hobby:char)

PK(idPersona,hobby)

FK1(idPersona)/Personas

FK2(hobby)/Hobbies

Resuelva las siguientes consultas en álgebra relacional y SQL

q1: ¿Lista de hobbies?

q2: ¿Lista de nombres de personas?

q3: ¿Nombres de personas y sus aficiones?

q4: ¿Personas a las que les gusta el Tenis?

q5: ¿Personas que no tienen hobbies?

q6: ¿Personas a quienes les gustan todos los hobbies?

q7: ¿Cuenta de aficiones?

q8: ¿Aficiones distintas?

q9: ¿Cuenta de aficiones de cada persona?

q10: ¿Máximo número de aficiones que tiene una persona?

q11: ¿Persona que tiene el máximo número de aficiones?

Solución Álgebra Relacional

q1: ¿Lista de hobbies?

$$Hobbies = \{Cine, Lectura, Fútbol, Tenis, Senderismo\};$$

q2: ¿Lista de nombres de personas?

$$\prod_{persona} Personas = \{Ana, Pedro, Joaquín, Luisa, Amorfo\};$$

q3: ¿Nombres de personas
y sus aficiones?

$$\prod_{persona, hobby} (Personas \bowtie Aficiones) = \left\{ \begin{array}{l} (Ana, Cine), \\ (Pedro, Lectura), (Pedro, Fútbol), \\ (Joaquín, Tenis), (Joaquín, Senderismo), \\ (Luisa, Cine), (Luisa, Lectura), (Luisa, Fútbol), (Luisa, Tenis), (Luisa, Senderismo), \end{array} \right\};$$

q4: ¿Personas a las que les gusta el Tenis?

$$\prod_{persona} \left(\sigma_{hobby='Tenis'} (Personas \bowtie Aficiones) \right) = \{Joaquín, Luisa\};$$

q5: ¿Personas que no tienen hobbies?

$$\prod_{persona} \left(Personas \bowtie \left(\left(\prod_{idPersona} Personas \right) - \left(\prod_{idPersona} Aficiones \right) \right) \right) = \{Amorfo\};$$

q6: ¿Personas a quienes les gustan todos los hobbies?

$$\frac{Aficiones}{Hobbies} \equiv \left(\prod_{idPersona} Aficiones \right) - \left(\prod_{idPersona} \left(\left(\left(\prod_{idPersona} Aficiones \right) \times Hobbies \right) - Aficiones \right) \right) = \{Luisa\}$$

q7: ¿Cuenta de aficiones?

$$\prod^{count(*)} Aficiones = \{10\};$$

q8: ¿Aficiones distintas?

$$\prod^{count(hobby)} \left(\prod_{hobby} Aficiones \right) = \{5\};$$

q9: ¿Cuenta de aficiones de cada persona?

$$\prod_{persona, count(*)} (Aficiones \bowtie Personas) = \{(Luisa, 5), (Joaquín, 2), (Pedro, 2), (Ana, 1)\};$$

q10: ¿Máximo número de aficiones que tiene una persona?

$$\prod^{max(na)} \left(\rho_{na / count(*)} \left(\prod_{idPersona}^{count(*)} Aficiones \right) \right) = \{5\};$$

q11: ¿Persona que tiene el máximo número de aficiones?

$$\prod^{max(na)} \left(\rho_{na \leftarrow count(*)} \left(\prod_{idPersona}^{count(*)} Aficiones \right) \right) = \prod^{count(*)} \left(\prod_{idPersona, persona}^{persona, count(*)} (Aficiones \bowtie Personas) \right) = \{Luisa\}$$

$$\{(5)\} \quad \{(5), (2), (2), (1)\} \quad \{(Luisa, 5), (Joaquín, 2), (Pedro, 2), (Ana, 1)\}$$

Solución SQL MariaDB

```

/* Crear la Base de Datos */
CREATE OR REPLACE DATABASE hobbies
  CHARACTER SET = utf8 COLLATE = 'utf8_general_ci';
USE hobbies;
/* Definición de tablas */
CREATE OR REPLACE TABLE personas (
  idPersona INTEGER KEY,
  persona VARCHAR(20) );
CREATE OR REPLACE TABLE hobbies (
  hobby VARCHAR(20) KEY
);
CREATE OR REPLACE TABLE aficiones (
  idPersona INTEGER,
  hobby VARCHAR(20),
  PRIMARY KEY(idPersona,hobby),
  FOREIGN KEY(idPersona)
    REFERENCES personas (idPersona),
  FOREIGN KEY(hobby) REFERENCES hobbies (hobby)
);

/* Escenario de prueba */
INSERT INTO Personas VALUES
  (1,'Ana'),
  (2,'Pedro'),
  (3,'Joaquín'),
  (4,'Luisa'),
  (5,'Amorfo');
INSERT INTO Hobbies VALUES
  ('Cine'),
  ('Lectura'),
  ('Fútbol'),
  ('Tenis'),
  ('Senderismo');
INSERT INTO Aficiones VALUES
  (1,'Cine'),
  (2,'Lectura'), (2,'Fútbol'),
  (3,'Tenis'), (3,'Senderismo'),
  (4,'Cine'), (4,'Lectura'),
  (4,'Fútbol'), (4,'Tenis'),
  (4,'Senderismo');

/* Queries */
-- q1: Lista de hobbies
SELECT * FROM Hobbies;

-- q2: Lista de nombres de personas
SELECT persona FROM Personas;

-- q3: Aficiones de cada persona. Ejemplos de join
SELECT persona,hobby FROM (Personas NATURAL JOIN Aficiones);
SELECT persona,hobby FROM Personas P, Aficiones A WHERE A.idPersona=P.idPersona;

-- q4: Personas a quién les gusta el Tenis
SELECT persona FROM Personas P
  WHERE EXISTS (SELECT * FROM Aficiones A
    WHERE A.idPersona = P.idPersona
    AND A.hobby = 'Tenis');

SELECT persona FROM (Personas NATURAL JOIN Aficiones) WHERE hobby = 'Tenis';

-- q5: Personas sin hobbies
SELECT persona FROM Personas P
  WHERE NOT EXISTS (SELECT * FROM Aficiones A
    WHERE A.idPersona=P.idPersona);

-- q6: Personas a las que les gusta todo
SELECT P.persona FROM Personas P
  WHERE NOT EXISTS (SELECT * FROM Hobbies H
    WHERE NOT EXISTS (SELECT * FROM aficiones A
      WHERE A.hobby = H.hobby
      AND A.idPersona=P.idPersona ));

SELECT persona FROM Personas NATURAL JOIN -- obtener nombre (persona)
  ( SELECT idPersona FROM Aficiones -- p que tienen todos los hobbies
  EXCEPT
  SELECT idPersona FROM -- p que no tienen algún hobby
    ( SELECT idPersona,hobby FROM Personas,Hobbies -- Todas las combinaciones p,h
      EXCEPT -- p,h que no tienen algún hobby
      SELECT idPersona,hobby FROM Aficiones -- p,h en hobbies
    ) X
  ) Y;

```

```
-- q7: Cuenta de aficiones que tienen las personas registradas
SELECT count(*) FROM Aficiones;

-- q8: Cuenta de aficiones distintas que existen registradas
SELECT COUNT(DISTINCT hobby) FROM Aficiones;

-- q9: Cuenta de aficiones de cada persona
SELECT persona, count(*) as na
  FROM (Personas NATURAL JOIN Aficiones)
  GROUP BY persona
  ORDER BY 2 DESC;

-- q10: Máximo número de aficiones que tiene una persona
SELECT max(na) FROM (SELECT COUNT(*) AS na FROM Aficiones GROUP BY idpersona) X;

-- q11: Persona que tiene el máximo número de aficiones
SELECT persona FROM (Personas NATURAL JOIN Aficiones) -- para extraer el nombre
  GROUP BY idpersona, persona -- query exterior con las cuentas de cada p
  HAVING COUNT(*) = ( SELECT MAX(na) FROM -- El máximo de todas las p
                      ( SELECT COUNT(*) AS na FROM Aficiones -- Cuenta para cada p
                        GROUP BY idpersona
                      ) X
                    );

SELECT persona FROM (Personas NATURAL JOIN Aficiones)
  GROUP BY idpersona, persona
  HAVING COUNT(*) = ( SELECT COUNT(*) AS na FROM Aficiones GROUP BY idpersona ORDER
BY 1 DESC LIMIT 1);

SELECT persona FROM (Personas NATURAL JOIN Aficiones)
  GROUP BY idpersona, persona
  HAVING COUNT(*) >= ALL ( SELECT count(*) FROM Aficiones
  GROUP BY idpersona );
```