

FBD

Examen VI



Los Del DGIIM, losdeldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

FBD

Examen VI

Los Del DGIIM, losdeldgiim.github.io

Arturo Olivares Martos

Granada, 2024-2025

Asignatura Fundamentos de Bases de Datos.

Curso Académico 2017-18.

Descripción Convocatoria Extraordinaria. Práctico Parcial 2 (Seminarios 3-4).

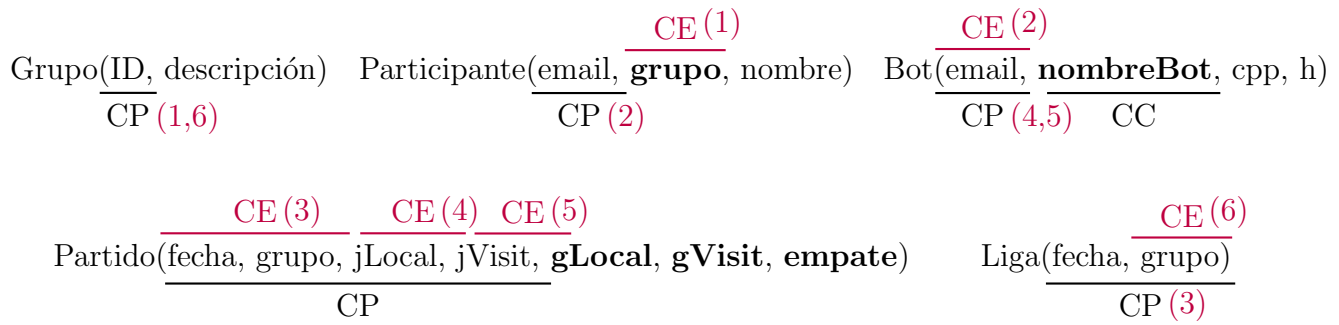


Figura 1: Diagrama Entidad-Relación

Disponemos de la BD de la Figura 1 sobre gestión de varias ligas de agentes inteligentes (bots). Hay varios grupos separados. Cada participante está en un grupo. Cada participante juega con un único bot. Hay varias ligas, identificadas por la fecha de comienzo de la liga y el grupo que juega la liga. En cada partido de liga de un grupo juegan dos bots diferentes (jLocal y jVisit) y se guarda si ha ganado el bot local (gLocal), el visitante (gVisit) o si hay empate (empate).

Ejercicio 1. Escribe las instrucciones en SQL para la creación de las tablas **Partido** y **Bot**. Se presuponen creadas todas las restantes tablas. Además de las restricciones de integridad especificadas en el dibujo, deben considerarse las siguientes:

1. Los campos en negrita no pueden ser nulos.
2. gLocal, gVisit y empate deben tener valores sólo 0 o 1 (gLocal=1 gana el local, gVisit=1 gana el visitante, empate=1 hay empate).
3. Solo es posible que gane un único jugador, o que haya empate.
4. Un bot no puede jugar consigo mismo un partido.
5. cpp y h deben ser cadenas largas, pues contendrán el código fuente del bot.

```

1 CREATE TABLE Bot (
    email CONSTRAINT Bot_PK PRIMARY KEY CONSTRAINT Bot_FK_Participante
    REFERENCES Participante(email),
    nombreBot VARCHAR2(50) CONSTRAINT Bot_NombreBot_NN NOT NULL CONSTRAINT
    Bot_NombreBot_UN UNIQUE,
    cpp VARCHAR2(4000),
5    h VARCHAR2(4000)
);
CREATE TABLE Partido (
    fecha,
    grupo,
10    jLocal CONSTRAINT Partido_FK_Bot_Local REFERENCES Bot(email),
    jVisit CONSTRAINT Partido_FK_Bot_Visit REFERENCES Bot(email),
    gLocal INT CONSTRAINT Partido_GLocal_CK CHECK (gLocal IN (0, 1)),
    gVisit INT CONSTRAINT Partido_GVisit_CK CHECK (gVisit IN (0, 1)),
    empate INT CONSTRAINT Partido_Empate_CK CHECK (empate IN (0, 1)),
15    CONSTRAINT Partido_PK PRIMARY KEY (fecha, grupo, jLocal, jVisit),
    CONSTRAINT Partido_FK_Liga FOREIGN KEY (fecha, grupo) REFERENCES Liga(fecha,
    grupo),

```

```

CONSTRAINT Partido_CK CHECK (gLocal + gVisit + empate = 1),
CONSTRAINT Partido_CK_Bot_Distinto CHECK (jLocal != jVisit)
);

```

Ejercicio 2 (AR y SQL). Mostrar los nombres de los participantes del grupo A1 cuyos bots han jugado algún partido como local en todas las ligas de su grupo.

Mostraremos en primer lugar el resultado en Álgebra Relacional. Las ligas del grupo A1 son:

$$\sigma_{\text{grupo}=\text{A1}}(\text{Liga})$$

Buscamos ahora todas las parejas participante del grupo A1, liga; en las que dicho participante ha jugado como local en algún partido de la liga:

$$\pi_{\substack{\text{email} \\ \text{fecha} \\ \text{grupo}}}(\sigma_{\text{grupo}=\text{A1}}(\text{Participante}) \bowtie_{\text{jLocal}=\text{email}} \text{Partido})$$

Por tanto, el email de los participantes del grupo A1 cuyos bots han jugado algún partido como local en todas las ligas de su grupo es:

$$\pi_{\substack{\text{email} \\ \text{fecha} \\ \text{grupo}}}(\sigma_{\text{grupo}=\text{A1}}(\text{Participante}) \bowtie_{\text{jLocal}=\text{email}} \text{Partido}) \div \sigma_{\text{grupo}=\text{A1}}(\text{Liga})$$

Por tanto, los nombres de los participantes del grupo A1 cuyos bots han jugado algún partido como local en todas las ligas de su grupo son:

$$\pi_{\text{nombre}} \left[\left[\pi_{\substack{\text{email} \\ \text{fecha} \\ \text{grupo}}}(\sigma_{\text{grupo}=\text{A1}}(\text{Participante}) \bowtie_{\text{jLocal}=\text{email}} \text{Partido}) \div \sigma_{\text{grupo}=\text{A1}}(\text{Liga}) \right] \bowtie \text{Participante} \right]$$

En SQL, la consulta sería:

```

1  SELECT nombre
   FROM Participante
  WHERE grupo = 'A1'
     AND NOT EXISTS (
5      SELECT *
       FROM Liga
      WHERE grupo = 'A1'
  MINUS
10     SELECT fecha, grupo
       FROM Partido
      WHERE jLocal = email
   );

```

Ejercicio 3 (SQL). Mostrar nombre y puntuación de aquellos bots que, jugando como local en la liga del 3 de Junio de 2018, obtuvieron un total de puntos superior a 4. La puntuación jugando como local se calcula como $3 \times \text{Partidos ganados como local} + \text{partidos empatados}$.

```
1 SELECT nombreBot, SUM(3*gLocal + empate) AS puntuacion
   FROM bot JOIN Partido ON email = jLocal
   WHERE TO_CHAR(fecha, 'DD-MM-YYYY') = TO_DATE('03-06-2018', 'DD-MM-YYYY')
   GROUP BY nombreBot
5  HAVING SUM(3*gLocal + empate) > 4;
```

Ejercicio 4 (AR y SQL). Mostrar los emails de los participantes que hayan ganado al bot de nombre **GreedyBot** como local y como visitante.

En Álgebra Relacional, los emails de los participantes que hayan ganado al bot de nombre **GreedyBot** como local y como visitante son:

$$\pi_{\text{email}}(\sigma_{\text{nombreBot}=\text{GreedyBot}}(\text{Bot}) \bowtie_{\text{email}=j\text{Local}} \sigma_{g\text{Local}=1}(\text{Partido})) \cap \\ \cap \pi_{\text{email}}(\sigma_{\text{nombreBot}=\text{GreedyBot}}(\text{Bot}) \bowtie_{\text{email}=j\text{Visit}} \sigma_{g\text{Visit}=1}(\text{Partido}))$$

En SQL, la consulta sería:

```
1 SELECT email
   FROM Bot JOIN Partido ON email = jLocal
   WHERE nombreBot = 'GreedyBot' AND gLocal = 1
INTERSECT
5 SELECT email
   FROM Bot JOIN Partido ON email = jVisit
   WHERE nombreBot = 'GreedyBot' AND gVisit = 1;
```