

FBD

Examen IV



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 IV

Los Del DGIIM, losdeldgiim.github.io

Arturo Olivares Martos

Granada, 2024-2025

Asignatura Fundamentos de Bases de Datos.

Curso Académico 2020-21.

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

1. Modelo 1

Considerar el esquema relacional de la Figura 1, donde cada paciente tiene asignada una aseguradora que cubre sus gastos médicos. La tabla **Consulta** almacena citas en las que un médico ha atendido a un paciente en una fecha dada.

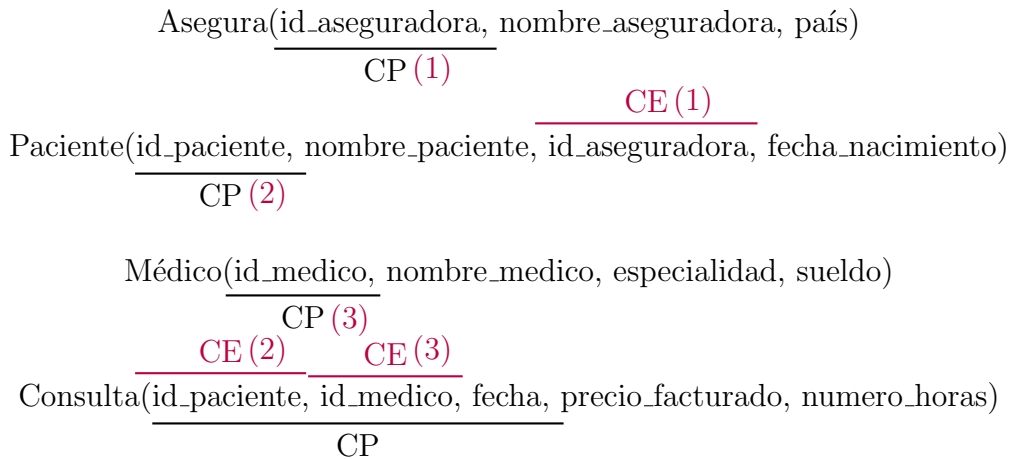


Figura 1: Esquema relacional del Modelo 1.

Ejercicio 1.1 (SQL). Encontrar, considerando sólo los médicos que han realizado consultas en el 2014, el sueldo del médico que ha realizado menos consultas durante dicho año. Puede haber más de un médico que cumpla esta condición.

```

1  SELECT id_medico, sueldo, count(*)
   FROM medico NATURAL JOIN consulta
   WHERE TO_CHAR(fecha, 'YYYY') = '2014'
   GROUP BY id_medico, sueldo
5  HAVING COUNT(*) = (
      SELECT MIN(COUNT(*))
      FROM consulta
      WHERE TO_CHAR(fecha, 'YYYY') = '2014'
      GROUP BY id_medico
10 );
  
```

Ejercicio 1.2 (SQL). Encontrar, entre los pacientes que han tenido consultas en el 2011, el id de la aseguradora de aquel paciente que ha gastado en total menos dinero sumando sus consultas en dicho año. Puede haber más de un paciente que cumpla esta condición.

En primer lugar, buscamos los pacientes que han tenido consultas en el 2011 y han gastado menos dinero sumando sus consultas en dicho año.

```

1  SELECT id_paciente, SUM(precio_facturado)
   FROM consulta
   WHERE TO_CHAR(fecha, 'YYYY') = '2011'
   GROUP BY id_paciente
5  HAVING SUM(precio_facturado) = (
      SELECT MIN(SUM(precio_facturado))
  
```

```
10      FROM consulta
        WHERE TO_CHAR(fecha, 'YYYY') = '2011'
        GROUP BY id_paciente
    );
```

Una vez obtenidos los pacientes que cumplen la condición, busquemos el id de la aseguradora de cada uno de ellos.

```
1  SELECT id_aseguradora, id_paciente, SUM(precio_facturado)
   FROM paciente NATURAL JOIN consulta
   WHERE TO_CHAR(fecha, 'YYYY') = '2011'
   GROUP BY id_aseguradora, id_paciente
5  HAVING SUM(precio_facturado) = (
        SELECT MIN(SUM(precio_facturado))
        FROM consulta
        WHERE TO_CHAR(fecha, 'YYYY') = '2011'
        GROUP BY id_paciente
10 );
```

Ejercicio 1.3 (SQL). Encontrar el nombre y sueldo de los médicos con especialidad VASCULAR que, para cada aseguradora que hay en la base de datos, han hecho al menos una consulta con un precio superior a 100 a un paciente de esa aseguradora.

Buscamos los médicos con especialidad VASCULAR para los que no existe una aseguradora para la que no hay una consulta con precio superior a 100 a un paciente de esa aseguradora.

```
1  SELECT id_medico, nombre_medico, sueldo
   FROM medico
   WHERE especialidad = 'VASCULAR'
      AND NOT EXISTS (
5      SELECT id_aseguradora
        FROM aseguradora
       MINUS
       SELECT id_aseguradora
        FROM paciente NATURAL JOIN consulta
10      WHERE precio_facturado > 100
          AND consulta.id_medico = medico.id_medico
    );
```

Ejercicio 1.4 (SQL). Encontrar el id_paciente y el id_aseguradora de los pacientes que, para cada especialidad de los médicos que tienen un sueldo superior a 4000, han sido atendidos al menos una vez durante menos de una hora por un médico con esa especialidad (gane lo que gane ese médico).

Buscamos los pacientes para los que no existe una especialidad de los médicos con sueldo superior a 4000 para la que no hay una consulta entre dicho paciente y un médico con esa especialidad que haya durado menos de una hora.

```

1  SELECT id_paciente, id_aseguradora
   FROM paciente
   WHERE NOT EXISTS (
       SELECT especialidad
5      FROM medico
       WHERE sueldo > 4000
   MINUS
       SELECT especialidad
10      FROM medico NATURAL JOIN consulta
       WHERE numero_horas < 1
       AND consulta.id_paciente = paciente.id_paciente
   );

```

Ejercicio 1.5 (SQL). Encontrar, considerando sólo los médicos que han realizado consultas en el año 2017, el nombre del médico que ha facturado en total más dinero sumando sus consultas durante dicho año. Puede haber más de un médico que cumpla esta condición.

```

1  SELECT id_medico, nombre_medico, SUM(precio_facturado)
   FROM medico NATURAL JOIN consulta
   WHERE TO_CHAR(fecha, 'YYYY')='2017'
   GROUP BY id_medico, nombre_medico
5  HAVING SUM(precio_facturado)=(
       SELECT MAX(SUM(precio_facturado))
       FROM consulta
       WHERE TO_CHAR(fecha, 'YYYY')='2017'
       GROUP BY id_medico
10 );

```

Ejercicio 1.6 (AR). Elaborar un listado con el id de los pacientes atendidos en consulta para los que todas sus consultas son siempre del mismo médico.

Creamos en primer lugar el siguiente alias.

$$\rho(\text{Consulta}) = C_1, C_2$$

Los pacientes buscados son los que no han sido atendidos por dos médicos distintos:

$$\pi_{\text{id_paciente}}(\text{Consulta}) - \pi_{C_1.\text{id_paciente}} \left(\sigma_{\substack{C_1.\text{id_paciente}=C_2.\text{id_paciente} \\ C_1.\text{id_médico} \neq C_2.\text{id_médico}}} (C_1 \times C_2) \right)$$

Ejercicio 1.7 (AR). Elaborar un listado de países con una única aseguradora. Creamos en primer lugar el siguiente alias.

$$\rho(\text{Aseguradora}) = A_1, A_2$$

Los pacientes buscados son los que no han sido atendidos por dos médicos distintos:

$$\pi_{\text{país}}(\text{Aseguradora}) - \pi_{A_1.\text{país}} \left(\sigma_{\substack{A_1.\text{país}=A_2.\text{país} \\ A_1.\text{id_aseguradora} \neq A_2.\text{id_aseguradora}}} (A_1 \times A_2) \right)$$

2. Modelo 2

Considerar el esquema relacional de la Figura 2, donde cada proyecto tiene asignada la empresa que lo elabora. La tabla **Revisión** registra revisiones indicando qué revisor revisa qué proyecto en qué fecha.

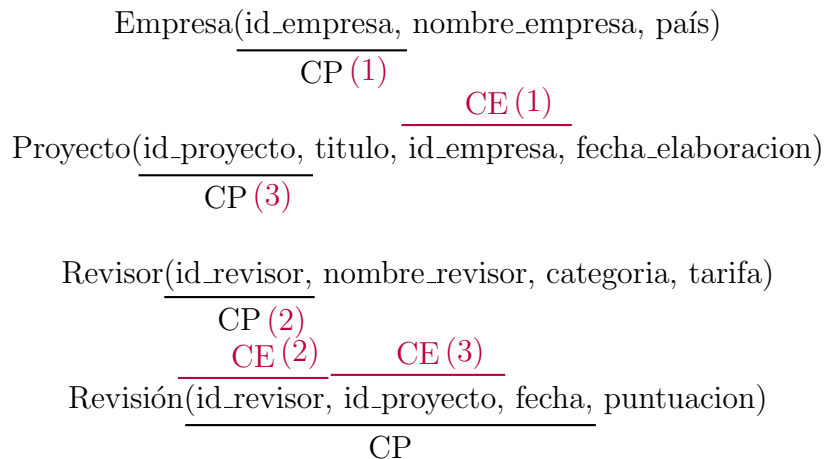


Figura 2: Esquema relacional del Modelo 2.

Ejercicio 2.1 (SQL). Encontrar, considerando sólo los revisores que han hecho revisiones en el 2012, el nombre del revisor que ha otorgado mayor total de puntos sumando todas sus revisiones de ese año. Puede haber más de un revisor que cumpla esta condición.

```

1  SELECT id_revisor, nombre_revisor, SUM(puntuacion)
   FROM revisor NATURAL JOIN revision
   WHERE TO_CHAR(fecha, 'YYYY') = '2012'
   GROUP BY id_revisor, nombre_revisor
5  HAVING SUM(puntuacion) = (
   SELECT MAX(SUM(puntuacion))
   FROM revision
   WHERE TO_CHAR(fecha, 'YYYY') = '2012'
   GROUP BY id_revisor
10 );
  
```

Ejercicio 2.2 (SQL). Encontrar el nombre y la tarifa de los revisores con categoría **SENIOR** que, para cada empresa que hay en la base de datos, han hecho al menos una revisión con una puntuación mayor de 75 puntos a un proyecto de esa empresa.

Buscamos los revisores con categoría **SENIOR** para los que no existe una empresa para la que no hay una revisión con puntuación mayor de 75 a un proyecto de esa empresa.

```

1  SELECT id_revisor, nombre_revisor, tarifa
   FROM revisor
   WHERE categoria = 'SENIOR'
   AND NOT EXISTS (
5   SELECT id_empresa
  
```

```
10      FROM empresa
      MINUS
      SELECT id_empresa
      FROM proyecto NATURAL JOIN revision
      WHERE puntuacion > 75
      AND revision.id_revisor = revisor.id_revisor
    );
```

Ejercicio 2.3 (SQL). Encontrar, considerando sólo proyectos que han tenido alguna revisión durante el año 2015, la fecha de elaboración del proyecto que ha tenido más revisiones en dicho año. Puede haber más de un proyecto que cumpla esta condición.

```
1  SELECT id_proyecto, fecha_elaboracion, COUNT(*)
   FROM proyecto NATURAL JOIN revision
   WHERE TO_CHAR(fecha, 'YYYY') = '2015'
   GROUP BY id_proyecto, fecha_elaboracion
5  HAVING COUNT(*) = (
      SELECT MAX(COUNT(*))
      FROM revision
      WHERE TO_CHAR(fecha, 'YYYY') = '2015'
      GROUP BY id_proyecto
10 );
```


3. Modelo 3

Considerar el esquema relacional de la Figura 3, donde cada vehículo tiene asignado un modelo de una marca determinada. La tabla **Repara** registra reparaciones indicando qué mecánico repara qué vehículo en qué fecha.

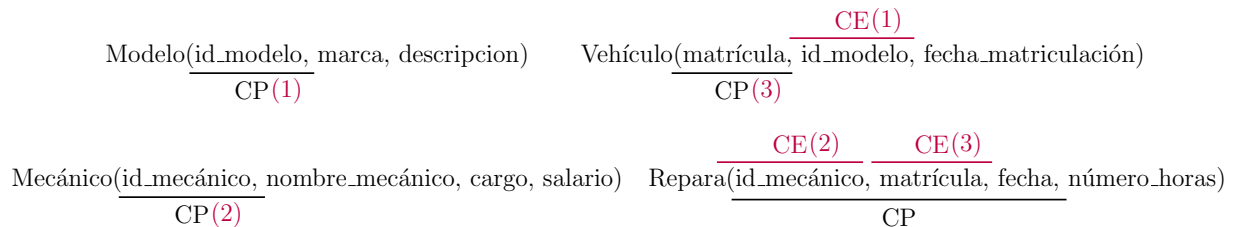


Figura 3: Esquema relacional del Modelo 3.

Ejercicio 3.1 (SQL). Encontrar, de entre los coches que han sufrido reparaciones en el año 2017, la fecha de matriculación del coche al que le han realizado más reparaciones en dicho año. Puede haber más de un coche que cumpla esta condición.

```

1  SELECT matrícula, fecha_matriculación, COUNT(*)
   FROM vehículo NATURAL JOIN repara
   WHERE TO_CHAR(fecha, 'YYYY') = '2017'
   GROUP BY matrícula, fecha_matriculación
5  HAVING COUNT(*) = (
      SELECT MAX(COUNT(*))
      FROM repara
      WHERE TO_CHAR(fecha, 'YYYY') = '2017'
      GROUP BY matrícula
10 );

```

Ejercicio 3.2 (SQL). Encontrar la matrícula y la fecha de matriculación de los vehículos con `id_modelo` igual a `AURIS` que, por cada cargo de mecánico que han en la BD, han sido atendidos al menos una vez durante más de una hora por un mecánico con ese cargo.

Buscamos los vehículos con `id_modelo` igual a `AURIS` para los que no existe un cargo de mecánico para el que no hay una reparación con duración mayor de una hora a un vehículo de ese modelo.

```

1  SELECT matrícula, fecha_matriculación
   FROM vehículo
   WHERE id_modelo = 'AURIS'
      AND NOT EXISTS (
5      SELECT cargo
      FROM mecánico
      MINUS
      SELECT cargo
      FROM mecánico NATURAL JOIN repara
10     WHERE número_horas > 1

```

```
);      AND repara.matrícula = vehículo.matrícula
```

4. Modelo 4

Considerar el esquema relacional de la Figura 4, donde cada grupo musical tiene asignado un género musical. La tabla **Actuación** registra actuaciones indicando qué grupo actúa en qué población y cuándo.

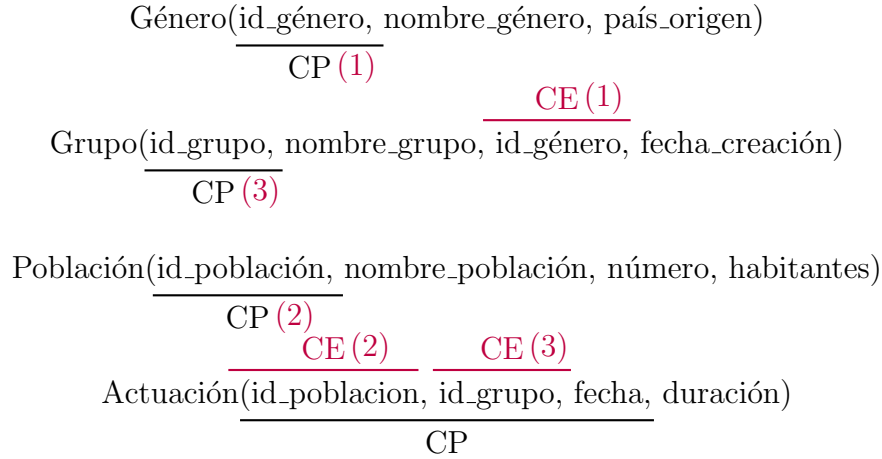


Figura 4: Esquema relacional del Modelo 4.

Ejercicio 4.1 (AR). Elaborar un listado de los países que han dado origen a un solo género musical.

Creamos en primer lugar el siguiente alias.

$$\rho(\text{Género}) = G_1, G_2$$

Los países buscados son los que no han dado origen a dos géneros musicales distintos:

$$\pi_{\text{país}}(\text{Género}) - \pi_{G_1.\text{país}} \left(\sigma_{\substack{G_1.\text{país}=G_2.\text{país} \\ G_1.\text{id.género} \neq G_2.\text{id.género}}} (G_1 \times G_2) \right)$$

5. Ejercicios Comunes

Ejercicio 5.1 (AR). Considere el esquema relacional de la Figura 5.

$$\begin{array}{ccc} \frac{A(F, B)}{\text{CP (1)}} & \frac{D(E, B)}{\text{CP (2)}} & \frac{(1) \overline{\text{CE}} \overline{\text{CE}} (2)}{\text{CP}} \end{array}$$

Figura 5: Esquema relacional del Ejercicio 5.1.

Encontrar los valores del atributo F de la tabla A que cumplen la condición de que aparecen relacionados en la tabla C con todos los valores de E que tienen asociado en la tabla D el mismo valor del atributo B que tiene asociado el valor de F en la tabla A . Es decir, un valor x debería aparecer en el resultado si existe una tupla $(x, y) \in A$ tal que, para cada tupla $(z, y) \in D$, existe una tupla $(x, z) \in C$ (siendo $x \in F$, $z \in E$ e $y \in B$).

Los valores del atributo F de la tabla A son:

$$\pi_F(A)$$

Los valores de E que tienen asociado en la tabla D el mismo valor del atributo B que tiene asociado el valor de F en la tabla A son:

$$\pi_E(D \bowtie A)$$

Por tanto, los valores del atributo F que aparecen relacionados en la tabla C con todos los valores de E que tienen asociado en la tabla D el mismo valor del atributo B que tiene asociado el valor de F en la tabla A son:

$$C \div \pi_E(D \bowtie A)$$

Como se pide además que dichos valores sean de A , la consulta pedida es:

$$\pi_F(A) \cap (C \div \pi_E(D \bowtie A))$$

No obstante, como F en C es una clave externa a A , se tiene que $\pi_F(C) \subset \pi_F(A)$, por lo que la consulta anterior es equivalente a:

$$C \div \pi_E(D \bowtie A)$$

Ejercicio 5.2 (AR). Considere el esquema relacional de la Figura 6.

$$\begin{array}{ccc} \frac{B(E, A)}{\text{CP (1)}} & \frac{D(F, A)}{\text{CP (2)}} & \frac{(1) \overline{\text{CE}} \overline{\text{CE}} (2)}{\text{CP}} \end{array}$$

Figura 6: Esquema relacional del Ejercicio 5.2.

Encontrar los valores del atributo E de la tabla B que cumplen la condición de que aparecen relacionados en la tabla C con todos los valores de F que tienen asociado en la tabla D el mismo valor del atributo A que tiene asociado el valor de E en la tabla B . Es decir, un valor x debería aparecer en el resultado si existe una tupla $(x, y) \in B$ tal que, para cada tupla $(z, y) \in D$, existe una tupla $(x, z) \in C$ (siendo $x \in E$, $z \in F$ e $y \in A$).

Los valores del atributo E de la tabla B son:

$$\pi_E(B)$$

Los valores de F que tienen asociado en la tabla D el mismo valor del atributo A que tiene asociado el valor de E en la tabla B son:

$$\pi_F(D \bowtie B)$$

Por tanto, los valores del atributo E que aparecen relacionados en la tabla C con todos los valores de F que tienen asociado en la tabla D el mismo valor del atributo A que tiene asociado el valor de E en la tabla B son:

$$C \div \pi_F(D \bowtie B)$$

Como se pide además que dichos valores sean de B , la consulta pedida es:

$$\pi_E(B) \cap (C \div \pi_F(D \bowtie B))$$

No obstante, como E en C es una clave externa a B , se tiene que $\pi_E(C) \subset \pi_E(B)$, por lo que la consulta anterior es equivalente a:

$$C \div \pi_F(D \bowtie B)$$