



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2413 - BASES DE DATOS

Interrogación 1

Fecha: 16 de abril de 2025

1° semestre 2025 - Profesores: Eduardo Bustos - Christian Álvarez

INSTRUCCIONES

- Al finalizar la prueba debe **digitalizarla y subirla a Canvas** correctamente. No seguir esta instrucción los expone a un descuento de hasta 5 décimas.
- Cada hoja de respuesta debe contener su **nombre completo y número de lista** en la parte superior.
- Responda cada pregunta en una hoja diferente.
- Debe firmar la lista de asistencia.
- Duración 2 horas.

Pregunta 1 - Álgebra Relacional

Sean las siguientes relaciones, que contienen información sobre **Jugadores**, **Rankings**, **Torneos** e **Inscripciones** de tenis. A continuación se muestran **ejemplos** de tuplas en las 4 tablas:

<u>id_jugador</u>	nombre	pais	edad
1	Novak Djokovic	Serbia	37
2	Carlos Alcaraz	España	21
3	Alejandro Tabilo	Chile	27
...
n	Luca Fernandini	Italia	23

Cuadro 1: Jugadores

<u>id_jugador</u>	<u>ranking_singles</u>	<u>ranking_dobles</u>
1	5	491
2	3	519
3	32	118
...
n	1	164

Cuadro 2: Rankings

<u>id_torneo</u>	<u>nombre_torneo</u>	<u>ciudad</u>	<u>superficie</u>	<u>puntos</u>
1	Wimbledon	Londres	pasto	2000
2	Roland Garros	Paris	arcilla	2000
3	Indian Wells	California	cemento	1000
...
m	Chile Open	Santiago	arcilla	250

Cuadro 3: Torneos

<u>id_inscripcion</u>	<u>id_jugador</u>	<u>id_torneo</u>	<u>fecha_inscripcion</u>
1	1	1	03-06-2025
2	3	1	12-06-2025
3	3	3	09-08-2025
...
k	n	m	31-01-2025

Cuadro 4: Inscripciones

Escriba las siguientes operaciones en álgebra relacional, utilizando la sintaxis vista en clases (puede encontrarla en el torpedito):

Hint: renombre las relaciones a J, R, T, I, respectivamente, con el operador correspondiente.

- a) (3 pts) Liste el nombre y país de los jugadores mayores de 35 años con un ranking en dobles menor a 50.

Solución:

Se renombran las relaciones por comodidad:

$$\rho(J, \text{Jugadores}), \quad \rho(R, \text{Rankings}), \quad \rho(T, \text{Torneos}), \quad \rho(I, \text{Inscripciones})$$

$$\pi_{\text{nombre,pais}} (\sigma_{\text{edad}>35 \wedge \text{ranking_dobles}<50} (J \bowtie_{J.id_jugador=R.id_jugador} R))$$

- b) (4 pts) Liste el nombre y fecha de inscripción de todos los jugadores que estén inscritos en torneos realizados en la ciudad de París.

Solución:

$$\pi_{\text{nombre, fecha_inscripcion}} (J \bowtie_{J.\text{id_jugador}=I.\text{id_jugador}} I \bowtie_{I.\text{id_torneo}=T.\text{id_torneo}} (\sigma_{\text{ciudad}='Paris'} T))$$

- c) (4 pts) Liste la ciudad de los torneos en los que ningún jugador chileno se haya inscrito.

Solución:

$$\begin{aligned} & \rho(\text{Chilenos}, \sigma_{\text{pais}='Chile'}(J)) \\ & \rho(\text{TorneosConChilenos}, \pi_{\text{id_torneo}} (\text{Chilenos} \bowtie_{\text{Chilenos.id_jugador}=I.\text{id_jugador}} I)) \\ & \pi_{\text{ciudad}}(T) - \pi_{\text{ciudad}}(T \bowtie_{T.\text{id_torneo}=\text{TorneosConChilenos.id_torneo}} \text{TorneosConChilenos}) \end{aligned}$$

- d) (4 pts) Liste el nombre, superficie y puntos de los torneos donde se han inscrito tanto jugadores mayores de 30 años como jugadores menores de 20 años.

Solución:

$$\begin{aligned} & \rho(T1, \pi_{\text{id_torneo}} ((\sigma_{\text{edad}>30} J) \bowtie_{J.\text{id_jugador}=I.\text{id_jugador}} I)) \\ & \rho(T2, \pi_{\text{id_torneo}} ((\sigma_{\text{edad}<20} J) \bowtie_{J.\text{id_jugador}=I.\text{id_jugador}} I)) \\ & \pi_{\text{nombre_torneo, superficie, puntos}} (T \bowtie_{T.\text{id_torneo}=(T1 \cap T2).\text{id_torneo}} (T1 \cap T2)) \end{aligned}$$

- e) (3 pts) **Bonus.** Liste el nombre y edad de los jugadores menores de 23 años que nunca hayan jugado un torneo en superficie de tipo *pasto*.

Solución:

$$\begin{aligned} & \rho(\text{Jovenes}, \sigma_{\text{edad}<23}(J)) \\ & \rho(\text{TPasto}, \sigma_{\text{superficie}='pasto'}(T)) \\ & \rho(\text{JugadoPasto}, \pi_{\text{nombre, edad}} (\text{Jovenes} \bowtie_{\text{Jovenes.id_jugador}=I.\text{id_jugador}} I \bowtie_{I.\text{id_torneo}=T.\text{id_torneo}} \text{TPasto})) \\ & \pi_{\text{nombre, edad}}(\text{Jovenes}) - \pi_{\text{nombre, edad}}(\text{JugadoPasto}) \end{aligned}$$

Pregunta 2 - Diagrama E/R y Esquema Relacional

Considere la red social de música DCCanciones (inspirada en Spotify), en la que los usuarios pueden seguir a otros usuarios y también interactuar con contenido musical.

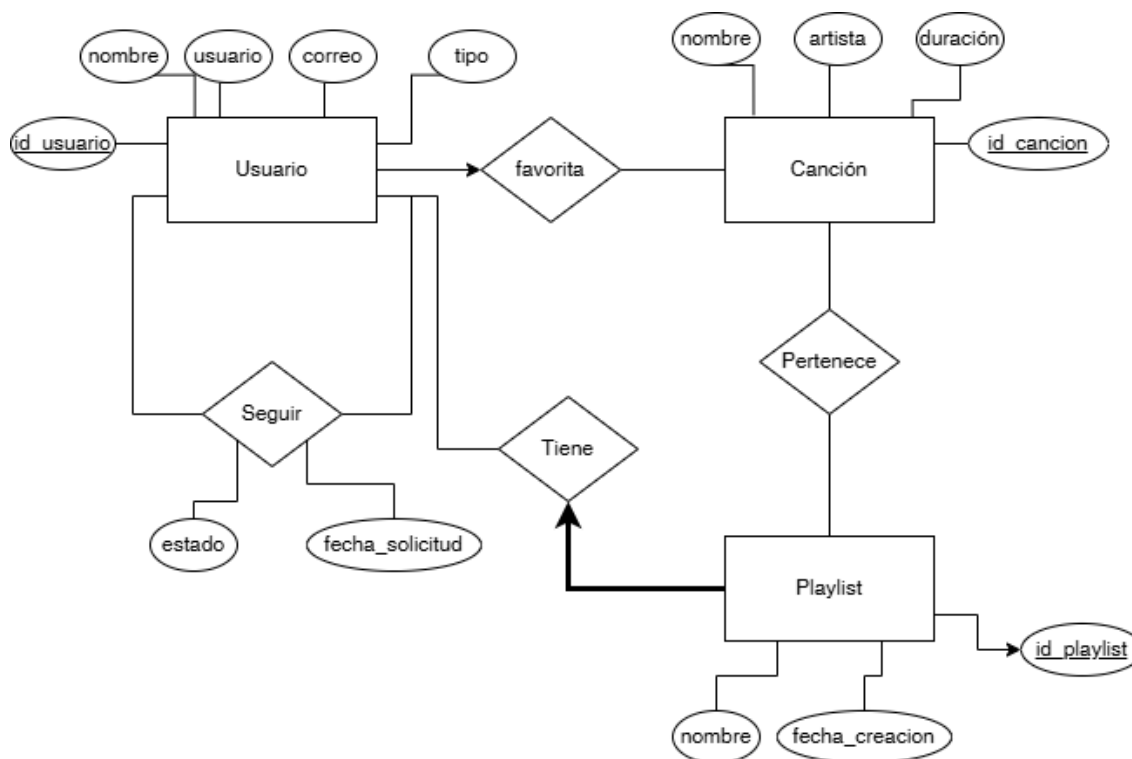
Cada usuario se identifica por un `id_usuario` y tiene atributos nombre, usuario, correo electrónico, canción favorita, y tipo. Los tipos de usuario pueden ser básico o premium. Los usuarios pueden seguir a otros usuarios dentro de la plataforma. Para esto, deben enviar una solicitud de seguimiento a otro usuario, la cual puede ser aceptada o rechazada posteriormente. Esta se registra con una fecha en la que se realizó la solicitud, y su estado actual.

Los usuarios pueden crear listas de reproducción (playlists), las que contienen canciones. Cada lista de reproducción tiene un nombre, una serie de canciones, una fecha de creación y está asociada a un único creador (usuario).

Las canciones están identificadas por un `id_cancion` y tienen atributos nombre, artista y duración. Una misma canción puede estar incluida en múltiples playlists.

- a) (10 pts) Diseñe el Diagrama Entidad/Relación de DCCanciones. Identifique claramente las entidades, entidades débiles, relaciones, cardinalidades, atributos, y llaves primarias y parciales.

Solución:



- b) (5 pts) A partir del Diagrama E/R propuesto, genere el Esquema Relacional asociado.

Para cada tabla, indique los atributos, sus dominios, las llaves primarias, foráneas, y cualquier otra aclaración respecto a la representación de sus relaciones.

Solución:

El esquema relacional asociado puede ser hecho de la manera vista en clases o bien utilizando sintaxis SQL. Ambas maneras son equivalentes.

```
CREATE TABLE Usuario (  
    id_usuario INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    usuario VARCHAR(30),  
    correo VARCHAR(100),  
    cancion_favorita INT,  
    tipo VARCHAR(10),  
    FOREIGN KEY (cancion_favorita) REFERENCES Cancion(id_cancion)  
);
```

```
CREATE TABLE Cancion (  
    id_cancion INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    artista VARCHAR(100),  
    duracion INT -- duración en segundos  
);
```

```
CREATE TABLE Playlist (  
    id_playlist INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    fecha_creacion DATE,  
    creador INT,  
    FOREIGN KEY (creador) REFERENCES Usuario(id_usuario)  
);
```

```
CREATE TABLE Seguir (  
    id_origen INT,  
    id_destino INT,  
    estado VARCHAR(10), -- pendiente, aceptada, rechazada  
    fecha_solicitud DATE,  
    PRIMARY KEY (id_origen, id_destino),  
    FOREIGN KEY (id_origen) REFERENCES Usuario(id_usuario),  
    FOREIGN KEY (id_destino) REFERENCES Usuario(id_usuario)  
);
```

```
CREATE TABLE Pertenece (  

```

```
    id_playlist INT,  
    id_cancion INT,  
    PRIMARY KEY (id_playlist, id_cancion),  
    FOREIGN KEY (id_playlist) REFERENCES Playlist(id_playlist),  
    FOREIGN KEY (id_cancion) REFERENCES Cancion(id_cancion)  
);
```

Pregunta 3 - SQL

Considere el siguiente esquema que tiene tablas que almacenan información sobre jugadores y equipos de fútbol:

- Jugador(nombre: string, equipo: string, origen: string, fecha_contratacion: date, calificacion: int, sueldo: bigint, posicion: ENUM(delantero, mediocampo, defensa, arquero))
- Partido(fecha: int, equipo_local: string, equipo_visita: string, golesLocal: smallint, golesvisita: smallint)
- Equipo(nombre: string, nombre_liga: string, pais_liga: string, division: int, continente: string)

En base a este esquema, construya las siguientes consultas utilizando SQL:

- a) (3 pts) El jugador que lleva más tiempo contratado entre todos los equipos. consulta simple

Solución:

```
SELECT nombre
FROM Jugador
WHERE fecha_contratacion = (SELECT MIN(fecha_contratacion) FROM Jugador);

-- Una manera alternativa válida:
SELECT nombre
FROM Jugador
ORDER BY fecha_contratacion ASC
LIMIT 1;
```

- b) (3 pts) Nombre del o los jugadores que llevan más tiempo contratados en cada equipo.

Solución:

```
SELECT
  j.nombre,
  j.equipo,
  j.fecha_contratacion
FROM
  Jugador j JOIN
  (SELECT equipo,
    MIN(fecha_contratacion) AS fecha_contratacion
  FROM Jugador
  GROUP BY equipo) AS subquery
```

```

ON j.equipo = subquery.equipo
   AND j.fecha_contratacion = subquery.fecha_contratacion;

-- Una manera alternativa válida

SELECT
  j.nombre,
  j.equipo,
  j.fecha_contratacion
FROM Jugador j,
  (SELECT equipo,
    MIN(fecha_contratacion) AS fecha_contratacion
  FROM Jugador
  GROUP BY equipo) AS subquery
WHERE j.equipo = subquery.equipo
   AND j.fecha_contratacion = subquery.fecha_contratacion;

```

- c) (3 pts) Todos los jugadores de equipos europeos cuyo origen es africano.

Solución:

```

SELECT Jugador.nombre, equipo.nombre    -- equipo.nombre es opcional
FROM Jugador, Equipo
WHERE Jugador.equipo = Equipo.nombre
   AND Equipo.continente = 'Europa'
   AND Jugador.origen = 'África';

-- Una manera alternativa válida:
SELECT Jugador.nombre
FROM Jugador
JOIN Equipo ON Jugador.equipo = Equipo.nombre
WHERE Equipo.continente = 'Europa' AND Jugador.origen = 'África';

```

- d) (3 pts) Ordene los equipos desde el mejor al peor, basado en el promedio de las calificaciones de sus jugadores.

Solución:

```

SELECT equipo, AVG(calificacion) AS calificacion_promedio
FROM Jugador
GROUP BY equipo
ORDER BY calificacion_promedio DESC;

```


- e) (3 pts) Obtenga el “dream team” con los mejores jugadores a nivel mundial en cada posición (para esto, considere 1 arquero, 4 defensas, 4 mediocampos y 2 delanteros).

Solución:

```
(SELECT nombre, equipo, calificacion, posicion
FROM Jugador
WHERE posicion = 'delantero'
ORDER BY calificacion DESC
LIMIT 2)
  UNION ALL
(SELECT nombre, equipo, calificacion, posicion
FROM Jugador
WHERE posicion = 'mediocampo'
ORDER BY calificacion DESC
LIMIT 4)
  UNION ALL
(SELECT nombre, equipo, calificacion, posicion
FROM Jugador
WHERE posicion = 'defensa'
ORDER BY calificacion DESC
LIMIT 4)
  UNION ALL
(SELECT nombre, equipo, calificacion, posicion
FROM Jugador
WHERE posicion = 'arquero'
ORDER BY calificacion DESC
LIMIT 1);
```

- f) (3 pts) **Bonus.** Obtenga la tabla de posiciones de los equipos de la liga *española*, ordenada de mayor a menor.

Solución:

```
SELECT equipo, SUM(puntos) AS total_puntos
FROM (
  SELECT equipo_local AS equipo,
  CASE
    WHEN golesLocal > golesvisita THEN 3
    WHEN golesLocal < golesvisita THEN 0
    ELSE 1
  END AS puntos
```

```

FROM Partido
WHERE equipo_local IN (SELECT nombre FROM Equipo WHERE pais_liga = 'España')

UNION ALL

SELECT equipo_visita AS equipo,
CASE
    WHEN golesLocal < golesvisita THEN 3
    WHEN golesLocal > golesvisita THEN 0
    ELSE 1
END AS puntos
FROM Partido
WHERE equipo_visita IN (SELECT nombre FROM Equipo WHERE pais_liga = 'España')
) AS Puntajes
GROUP BY equipo
ORDER BY total_puntos DESC;

-- Una alternativa válida utilizando partidos ganados para ordenar
SELECT equipo, COUNT(*) AS partidos_ganados
FROM (
    SELECT equipo_local AS equipo
    FROM Partido
    WHERE golesLocal > golesvisita
    AND equipo_local IN (SELECT nombre FROM Equipo WHERE pais_liga = 'España')

    UNION ALL

    SELECT equipo_visita AS equipo
    FROM Partido
    WHERE golesvisita > golesLocal
    AND equipo_visita IN (SELECT nombre FROM Equipo WHERE pais_liga = 'España')
) AS Ganados
GROUP BY equipo
ORDER BY partidos_ganados DESC;

```

Nota de corrección: Si el estudiante no sabe cómo se califica el puntaje, pueden contar solo partidos ganados, o bien goles convertidos. Si la sintaxis de CASE no la conocen, pueden proponer una sintaxis alternativa.

Pregunta 4 - Formas Normales

Usted ha sido contratado para analizar y rediseñar una base de datos utilizada por una empresa de transporte. Actualmente, la información sobre los envíos realizados por clientes se encuentra almacenada en una única tabla, como se muestra a continuación:

id_envío	cliente	rut_cliente	peso_kg	fecha	ciudad_entrega
1001	Sofía	15.815.461-7	10	2025-10-25	Santiago, Rancagua
1002	Paula	17.531.050-9	25	2025-11-06	Santiago, Talca, Curicó
1003	Ignacio	12.899.241-3	7	2025-03-17	Antofagasta, Calama
1004	Olivia	14.923.887-5	3	2025-01-21	Santiago, Temuco, Valdivia
1005	Consuelo	20.362.056-K	18	2025-09-08	Iquique, Arica
1006	Agustín	4.816.960-0	2	2025-08-30	Puerto Montt

Cuadro 5: Tabla Envíos

La columna `ciudad_entrega` contiene una lista con los nombres de las ciudades que forman parte del trayecto de entrega de cada envío. Un mismo envío puede pasar por una o más ciudades dependiendo de su destino, y el orden no importa.

A partir de la tabla anterior, responda lo siguiente:

- a) (5 pts) Identifique todas las dependencias funcionales presentes en la tabla.

Solución:

- `id_envío → cliente, rut_cliente, peso_kg, fecha, ciudad_entrega`
- `rut_cliente → cliente`

- b) (5 pts) Indique si la tabla se encuentra en 1NF, 2NF o 3NF. Justifique su respuesta.

Solución:

La tabla no se encuentra en 1NF, ya que el atributo `ciudad_entrega` contiene listas de ciudades, lo que viola el principio de atomicidad de la 1FN. Por lo tanto, tampoco cumple con 2NF ni 3NF.

- c) (5 pts) Proponga una descomposición del esquema que cumpla con BCNF, mostrando el nuevo conjunto de tablas y sus llaves primarias. Si no es posible, justifique por qué no es posible.

Solución:

Se propone la siguiente descomposición en BCNF:

- `Cliente(rut_cliente: string, cliente: string)`
Donde `rut_cliente → cliente`

-
- Envío(id_envío: int, rut_cliente: string, peso_kg: int, fecha: date)
Donde $\text{id_envío} \rightarrow \text{rut_cliente}, \text{peso_kg}, \text{fecha}$
 - EnvíoCiudad(id_envío: int, nombre_ciudad: string)
No hay dependencias funcionales no triviales fuera de la llave primaria.

En la tabla **Cliente**, el determinante de la única dependencia funcional es rut_cliente, el cual es llave primaria. En la tabla **Envío** ocurre lo mismo con la llave id_envío. En **EnvíoCiudad**, todos los atributos dependen únicamente de la llave compuesta (id_envío, nombre_ciudad). Por ende, la descomposición propuesta garantiza que toda dependencia funcional esté determinada por una llave candidata, lo que cumple con BCNF.

Nota de corrección: Se asume que el nombre de una ciudad es único. En caso de asumir lo contrario, basta con definir una nueva tabla que relacione el nombre de una ciudad con un id serial, e incluir el id en la tabla **EnvíoCiudad** en lugar del nombre de la ciudad.